

AD02/11

DIAGNOSTIC
MD-11-DZADA-C

EP-DZADA-C-DL-A

OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made In U.S.A.

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZADA-C
PRODUCT NAME:	AD02/AD11 DIAGNOSTIC TEST
DATE CREATED:	JUNE 12, 1974
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	EARL L. BOUSE

Vertical text on the left margin, possibly a page number or reference code.

1. ABSTRACT

THIS DIAGNOSTIC TESTS AND EXERCISES THE "AD02/AD11". THE PROGRAM IS SELF-STARTING AND WHEN LOADED WILL TYPE OUT THE PROGRAM TITLE AND THEN REQUEST THE A/D LENGTH. THE PROGRAM WILL ACCEPT A 10 TO 14 BIT UNIPOLAR OR BIPOLAR INPUT. EXAMPLE: 10(CR)* WOULD INDICATE A 10 BIT UNIPOLAR A/D; TYPING 11+(CR) WOULD INDICATE AN 11 BIT BIPOLAR A/D. A SENTENCE IS THEN TYPED GIVING THE LETTER DESIGNATORS TO BE TYPED TO RUN ANY ONE OF THE SIX (6) SEPERATE TESTS OF WHICH THIS PROGRAM IS COMPRISED. THE PROGRAM THEN TYPES A 'CR' AND THEN WAITS IN A KEYBOARD MONITOR MODE FOR A LETTER TO BE TYPED. ALTHOUGH THESE TESTS MAY BE RUN IN ANY ORDER IT IS IMPERATIVE THAT THE 'LOGIC' TEST IS RUN FIRST AND PROVED FULLY OPERATIONAL.

THE PROGRAM IS SET UP TO GIVE THE OPERATOR AS MUCH CONTROL OVER THE PROGRAM AS POSSIBLE VIA THE TELETYPE. TYPING A '*C' (OBTAINED VIA TYPING THE 'CTR' AND 'C' KEYS SIMULTANEOUSLY) WHILE RUNNING ANY TEST WILL ENABLE THE PROGRAM TO RETURN TO THE KEYBOARD MONITOR AND AWAIT A NEW LETTER DESIGNATOR TO BE TYPED. TYPING A '*A' WHILE IN MONITOR MODE WILL ENABLE THE LETTER DESIGNATORS TO BE RETYPED.

2. REQUIREMENTS (EQUIPMENT)

- A. PDP-11/05.15.20.45
- B. TELETYPE
- C. AD02/AD11 ANALOG TO DIGITAL CONVERTER

3. LOADING PROCEDURE

- A. USE STANDARD PROCEDURE FOR LOADING BINARY TAPES.

4. STARTING PROCEDURE

- A. THE PROGRAM IS SELF STARTING WITH A RESTART ADDRESS OF '200'.
- B. THE ABSOLUTE RESTART ADDRESS IS '174' IF A NEW A/D LENGTH IS TO BE ENTERED.

5. CONSOLE SWITCH SETTINGS

- A. ALL SWITCHES SHOULD BE DOWN (0) WHEN THE PROGRAM IS STARTED.
- B. REFER TO THE INDIVIDUAL TEST DESCRIPTIONS FOR APPLICABLE CONSOLE SWITCH SETTINGS

* TYPE 'CARRIAGE RETURN' (CR) TO TERMINATE ALL INPUT DATA.

6. AD02/AD11 ADDRESSES & FORMATS

A. A/D STATUS REGISTER (176770)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERROR				CH. ADDRESS				DONE			GAIN				

ALSO

- BIT 6 = INTERRUPT ENABLE
- BIT 2 = PRIORITY REQUEST
- BIT 1 = EXT. CLOCK ENABLE
- BIT 0 = A/D START

B. A/D BUFFER (176772)

1. VARIES WITH A/D LENGTH

7. TEST DIRECTORY

<u>TEST</u>	<u>SECTION</u>
LOGIC	8.
T.NNN	9.
CALIBRATION	10.
REPEATIBILITY	11.
GAIN ACCURACY	12.
RECOVERY	13.

8. LOGIC TEST

A. THE LOGIC TEST CONSISTS OF 39 SUBTESTS WHICH CHECK OUT THE 'AD02/AD11' LOGIC. EACH TEST IS LOOPED '2048' TIMES TO TEST LOGIC RELIABILITY.

B. RESTRICTIONS

DO 'NOT' CONNECT EXTERNAL SYNC FOR THE LOGIC TEST.

C. TEST TIME

IT TAKES APPROXIMATELY '1/2' A MINUTE TO RUN THE LOGIC TEST AND THE TELETYPE BELL IS RANG AT THE END OF EACH PASS.

Vertical text on the left margin, possibly a page number or reference code.

EO1

APR 20 11 21 AM '76
COMMUNICATIONS SECTION

MACY11 27(732) 16-SEP-76 16:37 PAGE 4

44
45

(8. CONT'D)

D. LOGIC ERRORS

ON ENCOUNTERING AN ERROR (DATA SWITCHES DOWN) THE ERROR ADDRESS AND THE CONTENTS OF THE A/D STATUS REGISTER ARE TYPED OUT.

1. ERROR FORMAT

ERROR ADDRESS A/D STATUS REGISTER

E. CONTROL SWITCHES (TELETYPE)

1. ^C (CONTROL C)

TYPING A ^C AT ANY TIME WILL ENABLE THE PROGRAM TO EXIT THE 'LOGIC' TEST AND RETURN TO THE MONITOR.

G. CONSOLE SWITCH SETTINGS FUNCTION

CONSOLE SW11=0	NORMAL RUN (2048 PASSES/TEST)
CONSOLE SW11=1	SUPPRESS SUBPROGRAM ITERATIONS
CONSOLE SW13=0	PRINT ERROR MESSAGE
CONSOLE SW13=1	INHIBIT ERROR MESSAGES
CONSOLE SW14=1	RUN SCOPE MODE
CONSOLE SW14=0	INHIBIT SCOPE MODE
CONSOLE SW15=0	CONTINUE AFTER TYPING ERROR
CONSOLE SW15=1	HALT ON ERROR

9. T'NNN (LOGIC TEST AID)

A. THIS ROUTINE IS DESIGNED TO ALLOW THE OPERATOR TO LOOP ON ANY 'LOGIC SUBTEST' REGARDLESS IF THE TEST FAILS OR NOT.

B. STARTING SEQUENCE

TYPE 'TNNN (CR)' WHERE 'NNN' IS THE OCTAL ADDRESS OF THE 'TESTX' TO BE EXECUTED. THE PROGRAM WILL THEN EXECUTE THE SUBTEST AND WILL REMAIN IN A SCOPE LOOP UNTIL THE COMPUTER IS STOPPED OR A '^C' IS TYPED TO RETURN TO THE MONITOR.

C. RESTRICTIONS

SWITCH '11' MUST BE '0' (DOWN) TO RUN THIS TEST.

146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

GO1

AD02 AD:1 DIAGNOSTIC
DEADAC.P11

MACY11 27(732) 16-SEP-76 16:37 PAGE 6

EO2
EO3

204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260

10. A/D CALIBRATION TEST

A. THE 'A/D CALIBRATION' TEST IS DESIGNED TO ACCEPT AN INPUT FROM THE TELETYPE TO INDICATE THE TYPE OF SYNC (EXTERNAL OR INTERNAL TO BE USED AND THEN TAKES CONTINUOUS CONVERSIONS USING THE 'CH.' AND 'GAIN' SELECTED VIA THE CONSOLE SWITCHES. THESE SETTINGS MAY BE CHANGED AT ANY TIME. THE CONVERSION VALUE IS EITHER PLACED IN RO AND DISPLAYED IN THE DATA LIGHTS VIA ISSUING RESETS OR PRINTED OUT ON THE TELEPRINTER.

B. STARTING SEQUENCE

1. TYPE 'C' TO RUN THE A/D CALIBRATION TEST.
2. THE TEST HEADER PLUS A REQUEST FOR A SYNC TYPE WILL THEN BE TYPED.
3. TYPE IN THE DESIRED SYNC (CR), 'I' FOR INTERNAL; 'E' FOR EXTERNAL.
4. THE PROGRAM WILL RESPOND VIA TYPING A CARRIAGE RETURN-LINE FEED AND THE TEST WILL START.

C. CALIBRATION ERROR

1. THE PRINTOUT 'ERROR BIT SET' WILL OCCUR WHILE RUNNING WITH EXTERNAL SYNC IF THE SYNC FREQUENCY IS TOO FAST.

D. CONTROL SWITCHES (TELETYPE)

1. ↑A (CONTROL A)

TYPING ↑A WILL ENABLE A NEW SYNC TYPE TO BE ENTERED.

2. ↑C (CONTROL C)

TYPING ↑C WILL CAUSE THE PROGRAM TO EXIT THE CALIBRATION TEST AND RETURN TO THE MONITOR.

E. CONSOLE SWITCH SETTINGS FUNCTION

SWITCHES '0-2'	CONVERSION DELAY
SWITCHES '3-4'	GAIN SELECT (1-8)
SWITCH '7=0'	DISPLAY VALUE IN RO
SWITCH '7=1'	PRINT CONVERSION VALUE
SWITCHES '9-15'	CHANNEL SELECT

11. REPEATIBILITY TEST

- A. THIS TEST REQUESTS A CH.(S), GAIN AND A COUNT SPREAD OF '1-4' TO BE TYPED IN BY THE OPERATOR. A SERIES OF '512' CONVERSIONS ARE THEN TAKEN ON THE INPUT CH.(S) AT THE SELECTED GAIN. CONVERSIONS ARE THEN AVERAGED OUT AND IF THE COUNT SPREAD IS FOUND TO BE GREATER THAN REQUEST, THE RESULTS OF THE CONVERSIONS ARE TYPED OUT. A SINGLE CHANNEL OR A SERIES OF CHANNELS MAY BE TESTED VIA TYPING EITHER 'N(CR)*' TO SELECT A SINGLE CHANNEL OR 'N,N(CR)' TO TEST A SERIES OF CHANNELS.

*NOTE 'N' WOULD BE A DECIMAL NO.

B. RESTRICTIONS

1. IF A SECOND CH. IS ENTERED, IT MUST BE LARGER THAN THE FIRST CH. OR THE INPUT WILL NOT BE ACCEPTED.
2. A VOLTAGE MUST BE PROVIDED TO THE SELECTED CH. WHICH IS LESS THAN FULL SCALE FOR THE SELECTED GAIN.

C. STARTING SEQUENCE

1. TYPE 'R' TO RUN THE 'REPEATIBILITY' TEST.
2. A REQUEST IS THEN MADE FOR CH.(S) TO BE TESTED, GAIN AND COUNT SPREAD (RANGE IN WHICH ALL 512 COUNTS MUST FALL FOR THE CH. TO BE CONSIDERED ACCEPTABLE).
3. IF THE CHANNEL IS FOUND TO BE WITHIN THE SELECTED COUNT SPREAD, THE PROGRAM WILL EITHER CONTINUE TO THE NEXT CHANNEL IF SELECTED OR RETEST THE CURRENT CHANNEL.

D. CONTROL SWITCHES (TELETYPE)
-----1. ↑A (CONTROL A)

TYPING A ↑A WHILE THE PROGRAM IS RUNNING WILL ENABLE A NEW CH.(S), GAIN AND COUNT SPREAD TO BE SELECTED.

2. ↑C (CONTROL)

TYPING ↑C WILL CAUSE THE PROGRAM TO EXIT THE 'REPEATIBILITY' TEST AND RETURN TO THE MONITOR.

258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308

309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362

E.	<u>CONSOLE SWITCH SETTINGS</u>	<u>FUNCTION</u>
	CONSOLE SW 10=0	NORMAL RUN
	CONSOLE SW 10=1	PRINTOUT ALL CONVERSIONS
	CONSOLE SW 13=0	PRINT ERRORS
	CONSOLE SW 13=1	INHIBIT ERROR PRINTOUTS

F. REPEATIBILITY ERRORS

ON ENCOUNTERING AN ERROR (CONSOLE SWITCHES DOWN) THE ERROR DATA IS TYPED OUT.

1. ERROR FORMAT

CH.	LO	AV	HI									
A	B	C	D									
LO	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	HI
E	F	G	H	I	J	K	L	M	N	O	P	Q

WHERE:
 A=CHANNEL BEING TESTED
 B=THE LOWEST READING OF THE '512' CONVERSIONS
 C=THE AVERAGE READING OF THE '512' CONVERSIONS
 D=THE HIGHEST READING OF THE '512' CONVERSIONS
 E=NUMBER OF COUNTS 'OUT OF RANGE' LOWER THAN 5 COUNTS
 F-J=NUMBER OF COUNTS IN EACH PART LOWER THAN AVERAGE.
 K=NUMBER OF COUNTS AT AVERAGE OF THE '512'
 L-P=NUMBER OF COUNTS IN EACH PART HIGHER THAN AVERAGE.
 Q=NUMBER OF COUNTS 'OUT OF RANGE' HIGHER THAN 5 COUNTS

12. GAIN TEST

A. THE GAIN TEST IS USED TO DETERMINE THE ACCURACY OF THE 'AD02/AD11' AT DIFFERENT GAIN SETTINGS. THE TEST REQUESTS 16 SPECIFIC VOLTAGES (8 FOR A UNIPOLAR A/D) TO BE APPLIED TO THE SELECTED CHANNEL. A SERIES OF '512' CONVERSIONS ARE TAKEN FOR EVERY VOLTAGE AND APPLICABLE GAIN SETTINGS AND THE AVERAGE IS COMPARED AGAINST THE TRUE VALUE FOR THAT SPECIFIC SETTING. IF THE AVERAGE IS MORE THAN + OR -1 COUNT FROM THE TRUE VALUE IT IS CONSIDERED IN ERROR AND THE CONVERSION RESULTS ARE TYPED OUT. AFTER TESTING ALL THE VOLTAGES AT THE SPECIFIED GAIN SETTINGS A TABLE OF THE RESULTS ARE TYPED OUT. WHEN THE COMPLETE TABLE HAS BEEN TYPED THE PROGRAM WILL REQUEST A NEW CH. TO BE TESTED.

363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418

B. STARTING PROCEDURE

1. TYPE 'G' TO RUN GAIN TEST.
2. THE MESSAGE "GAIN ACCURACY TEST. SUPPLY THE FOLLOWING VOLTAGES TO THE SELECTED CH., TYPE 'CR' TO START TEST."
3. A CH. AND A SPECIFIC VOLTAGE IS THEN REQUESTED.
4. TYPE 'CR' AFTER SUPPLYING THE REQUESTED VOLTAGE.

C. CONTROL SWITCHES (TELETYPE)

1. ↑A (CONTROL A)
TYPING A '↑A' WILL ENABLE THE GAIN TEST TO BE RESTARTED.
2. ↑C (CONTROL C)
TYPE A '↑C' TO RETURN CONTROL TO THE MONITOR.

D. CONSOLE SWITCH SETTINGS FUNCTION

CONSOLE SW13=0	PRINT GAIN ERROR
CONSOLE SW13=1	INHIBIT TYPEOUT
CONSOLE SW14=0	LOOP ON GAIN ERROR
CONSOLE SW14=1	INHIBIT ERROR LOOPING

E. GAIN ERRORS

ON ENCOUNTERING AN ERROR (CONSOLE SWITCHES SET TO '0') THE ERROR HEADER AND ERROR DATA IS TYPED OUT AND THE TEST IS LOOPED UNTIL EITHER THE CORRECT CONVERSION RESULTS ARE OBTAINED OR SW14 IS SET TO A '1' ALLOWING THE TEST TO CONTINUE.

1. ERROR FORMAT

GAIN	VOLTAGE	AVERAGE
A	B	C

WHERE:

A=GAIN SETTING
 B=TRUE VOLTAGE VALUE
 C=AVERAGE OF THE CONVERSION

2. GAIN CONVERSION TABLE (EXAMPLE OF AN '11' BIT BIPOLAR A/D)

GAIN	5.0000	2.5000	1.2500	0.6250	0.3125	0.1563	0.0781	0.0390
1	2000	1000	400	200	100			
2	-----	2000	1000	400	200	100		
4	-----	-----	2000	1000	400	200	100	
8	-----	-----	-----	2000	1000	400	200	100
16	776000	777000	777400	777600	777700			
32	-----	776000	777000	777400	777600	777700		
64	-----	-----	776000	777000	777400	777600	777700	

LO1

AD02/AD11 DIAGNOSTIC
DZADAC.P11

MACY11 27(732) 16-SEP-76 16:37 PAGE 11

419
420
421

8 ----- 776000 777000 777400 777600 777700

-7-

470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577

13. RECOVERY TEST

A. THE "RECOVERY TEST" IS DESIGNED TO DETERMINE THE RECOVERY CAPABILITY OF THE 'AD02/AD11' GAIN AMPLIFIERS. THE TEST REQUESTS FOR TWO (2) CH. AND TWO GAIN INPUTS TO BE TYPED IN. THE TEST THEN TAKES A SERIES OF SIXTEEN (16) CONVERSIONS (8 ON EACH CH.) AND THEN TYPES OUT THE '8' CONVERSION VALUES IN THE ORDER THEY WERE TAKEN ON THE SECOND CH.

B. STARTING SEQUENCE

1. TYPE 'E' TO RUN THE RECOVERY TEST.
2. A REQUEST IS THEN MADE FOR THE CH.S TO BE TESTED.
3. TYPE 'N,N (CR)' WHERE 'N' IS ANY DECIMAL CH.
4. A REQUEST FOR 'GAINS' IS THEN MADE.
5. TYPE 'N,N (CR)' WHERE 'N' WILL BE '1,2,4 OR 8.'
6. THE PROGRAM WILL THEN TAKE CONTINUOUS CONVERSIONS TYPING OUT THE CONVERSION VALUES FOR THE SECOND CH.

EXAMPLE:

CH. A XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX

WHERE:

A=TO THE SECOND CH.
X=TO THE '8' CONVERSIONS TAKEN ON THAT CH.

C. CONTROL SWITCHES (TELETYPE)

1. ↑A (CONTROL A)
TYPING A '↑A' WILL ENABLE A NEW SET OF CH.S AND GAINS TO BE ENTERED.
2. ↑C (CONTROL C)
TYPING A '↑C' WILL ENABLE THE PROGRAM TO RETURN TO THE MONITOR.

D. CONSOLE SWITCH SETTINGS FUNCTION

CONSOLE SW 13=0	PRINT CONVERSION VALUES
CONSOLE SW 13=1	INHIBIT PRINTOUT

14. LISTING

-8-

```
%
      .TITLE AD02/AD11 DIAGNOSTIC
      .ABS
;MAINDEC-11-DZADA-C
;COPYRIGHT 1972, 1974
;DIGITAL EQUIPMENT CORP. MAYNARD MASS. 01754
```



```

478
479
480
481
482      100000
483      040000
484      020000
485      010000
486      004000
487      002000
488      001000
489      000400
490      000200
491      000100
492      000040
493      000020
494      000010
495      000004
496      000002
497      000001
498
499
500
501      000000
502      000001
503      000002
504      000003
505      000004
506      000005
507      000006
508      000007
509
510
511
512      000000
513      000010
514      000020
515      000030
516
517
518      000001
519      000004
520      005746
521      005726
522      010046
523      012600
524      024646
525      022626
526      000240

```

;PROGRAMMER: EARL L. BOUSE

;SWITCH REGISTER DEFINITIONS AND FUNCTIONS:

```

SW15=100000      ;=1, HALT ON ERROR
SW14=40000       ;=1, LOOP ON CURRENT TEST
SW13=20000       ;=1, SUPPRESS ERROR TYPEOUT
SW12=10000
SW11=4000        ;=1, SUPPRESS 'SUBPROGRAM' ITERATIONS
SW10=2000        ;=1, FORCE TYPEOUT (REPEATIBILITY)
SW09=1000
SW08=400
SW07=200
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1

```

;REGISTER DEFINITIONS

```

R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7

```

;GAIN EQUIVALENCE TABLE

```

G1=00
G2=10
G4=20
G8=30

```

;INSTRUCTIONS DEFINITIONS

```

ADCSR=%1
INTVC=%4
PUSH1SP=5746
POP1SP=5726
PUSHR0=10046
POPR0=12600
PUSH2SP=24646
POP2SP=22626
NOP=240

```

```

;AD STATUS REGISTER
;INTERRUPT VECTOR

```


000024	011534	
000026	000340	
000060	011330	
000062	000340	
000030	000030	
000032	000340	
000034	011242	
000036	000340	
000174	000174	000600
000200	000167	001064

. = 24	
POWER FAIL	
340	
. = 60	
XTTYIN	
340	
. = 30	
EMTSRV	
340	
LOGERR	
340	
. = 174	
JMP	INIT
JMP	MONITR

:POWER FAIL HANDLER

:TELEPRINTER KEYBOARD ROUTINE

:EMT TRAP, EMT DISPATCH SERVICE

:TRAP TRAP, LOGIC ERROR TRAP

:INITIALIZATION ADDRESS

:PROGRAM 'RESTART' ADDRESS

:TRAP EQUIVALENCE TABLE:

104400

104000

104001

104002

104003

104004

104005

104006

104007

104010

104011

104012

104013

104014

104015

104016

104017

ERROR=TRAP

PRINT=EMT

DECOCT=EMT+1

SCOPE=EMT+2

GAININ=EMT+3

CMPUTE=EMT+4

CATORIZ=EMT+5

BINDEC=EMT+6

SPACE=EMT+7

PRTOCT=EMT+10

TTYIN=EMT+11

WAITGN=EMT+12

TAKEGN=EMT+13

PRTAGV=EMT+14

SIXOSH=EMT+15

TESTKS=EMT+16

GAINXN=EMT+17

:LOGIC TEST ERROR ROUTINE

:MESSAGE PRINTER ROUTINE

:DECIMAL TO OCTAL CONVERSION ROUTINE

:LOGIC TEST SCOPE SUBROUTINE

:ROUTINE TO REQUEST GAIN FROM TTY

:A/D AVERAGING ROUTINE

:ROUTINE TO CALCULATE THE COUNT SPREAD

:BINARY TO DECIMAL CONVERSION ROUTINE

:TYPE 'N' SPACES

:OCTAL PRINT ROUTINE

:TELETYPE INPUT ROUTINE

:GAIN TEST CONVERSION ROUTINE

:GAIN TEST CONVERSION ROUTINE

:GAIN AVERAGE PRINT ROUTINE

:SUBROUTINE TO TYPE OUT '6' DASHES

:SUBROUTINE TO TEST FOR KEYBOARD FLAG

:SUBROUTINE TO DECODE A GAIN FROM TTY

:TEST INITIALIZATIN ROUTINE. PROGRAM IS SELF STARTING TO THIS ROUTINE.
:THE ROUTINE IS EXECUTED ON LOADING ONLY

569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624

001000
001000 016706 014102
001004 012777 000340 014034
001012 005737 000042
001016 001402
001020 000167 000512
001024 012777 000100 014016
001032 104000
001034 013423
001036 005067 014034
001042 104000
001044 013504
001046 104011
001050 104001
001052 022767 000012 010356
001060 001004
001062 012767 001000 014022
001070 000442
001072 022767 000013 010336
001100 001004
001102 012767 002000 014002
001110 000432
001112 022767 000014 010316
001120 001004
001122 012767 004000 013762
001130 000422
001132 022767 000015 010276
001140 001004
001142 012767 010000 013742
001150 000412
001152 022767 000016 010256
001160 001403
001162 104000
001164 014052
001166 000723
001170 012767 020000 013714
001176 012700 015240
001202 016767 013704 013722
001210 016720 013716
001214 006267 013712
001220 022700 015252
001224 001371
001226 016767 013660 013676
001234 005467 013672
001240 016720 013666
001244 006267 013662
001250 022700 015264
001254 001371
001256 006367 013630
001262 104000
001264 013662

INIT: .=1000
 MOV STACK,SP
 MOV #340,SPSW
 TST #42
 BEQ +6
 JMP RESTR
 MOV #100,RTKS
 PRINT
 TITLE
INIT1: CLR ADSIGN
 PRINT
 MES2
 TTYIN
 DECOCT
 CMP #12,BCDTAB
 BNE SIZE11
 MOV #1000,ADSIZE
 BR LDSIZE
SIZE11: CMP #13,BCDTAB
 BNE SIZE12
 MOV #2000,ADSIZE
 BR LDSIZE
SIZE12: CMP #14,BCDTAB
 BNE SIZE13
 MOV #4000,ADSIZE
 BR LDSIZE
SIZE13: CMP #15,BCDTAB
 BNE SIZE14
 MOV #10000,ADSIZE
 BR LDSIZE
SIZE14: CMP #16,BCDTAB
 BEQ SIZE14A
 PRINT
 QMARK
 BR
SIZE14A: MOV #20000,ADSIZE
 LDSIZE: MOV #POS500,RO
 MOV ADSIZE,TEMP1
LDPOS: MOV TEMP1,(RO)+
 ASR TEMP1
 CMP #NEG500,RO
 BNE LDPOS
 MOV ADSIZE,TEMP1
LDNEG: MOV TEMP1,(RO)+
 ASR TEMP1
 CMP #NEG312+2,RO
 BNE LDNEG
 ASL ADSIZE
INITA: PRINT
 MES4

:INIT STACK POINTER=100C
:TEST FOR 'DDP1' MONITOR
:CONTINUE IF NOT OUT THERE
:IF PRESENT, RUN LOGIC TEST
:ENABLE TTY INTERRUPTS
:CALL MESSAGE PRINTER VIA 'EM'
:TYPE PROGRAM HEADER.
:BIPOLAR=C,UNIPOLAR=1
:REQUEST THE A/D LENGTH
:WAIT FOR ENTRY
:CONVERT A/D LENGTH TO OCTAL
:A/D LENGTH=TO 10 BITS?
:BRANCH IF NO
:YES, LOAD +5V VALUE
:BRANCH AND SET UP TABLE
:A/D LENGTH=10 11 BITS?
:BRANCH IF NOT =
:YES, LOAD +5V VALUE
:A/D LENGTH=TO 12 BITS?
:BRANCH IF NOT =
:YES, LOAD +5V VALUE
:A/D LENGTH=TO 13 BITS?
:LOAD 5 VOLT VAL
:TEST FOR 14 BITS
:RETRY
:LOAD +5V VALUE
:LOADED ALL POS. VALUES?
:BRANCH IF NO.
:SET UP NEG. VALUES
:LOADED ALL NEG VALUES?
:BRANCH IF NO
:SET UP OFFSET FOR AVERAGING ROUTINE
:PRINT THE TEST CALL LETTERS.

E02

AD02/AD11 DIAGNOSTIC MACY11 27(732) 16-SEP-76 16:37 PAGE 17
OZADAC.P11

625 001266 000420

RR INIT2

;GO AND AWAIT COMMAND.


```

626                                     ;MONITOR SUBROUTINE. ENTER VIA 'IC' OR A RESTART AT LOCATION '200'.
627
628 001270 000005 MONITR: RESET ;INITIALIZE ON ENTRY
629 001272 016706 013610 MOV STACK,SP ;RESET STACK POINTER
630 001276 012777 000340 013542 MOV #340,SPSW
631 001304 005737 000042 TST #42 ;TEST FOR 'DDPI' MONITOR
632 001310 001402 BEQ .+6 ;CONTINUE IF NOT OUT THERE
633 001312 000167 000220 JMP RESTR ;IF PRESENT, RUN LOGIC TEST
634 001316 012777 000100 013524 MOV #100,ATKS ;ENABLE TTY INTERRUPTS
635 001324 104000 PRINT ;CALL MESSAGE PRINTER
636 001326 014036 CNTRL C ;TYPE 'IC'
637
638 001330 012767 001262 013552 INIT2: MOV #IN,TA,AVECTR ;SET UP 'TA' VECTOR ADDRESS.
639 001336 012777 000100 013504 MOV #100,ATKS ;ENABLE KEYBOARD INTERRUPT
640 001344 005067 013530 CLR SUBX ;CLR SUBTEST REG. SWITCH
641 001350 104000 PRINT
642 001352 014047 DOT ;PRINT '.' TO INDICATE MONITOR READY
643 001354 104011 TTYIN ;WAIT FOR TTY ENTRY
644 001356 122767 000114 007706 CMPB #114,INBUF ;TEST FOR 'L'
645 001364 001002 BNE .+6 ;NOT 'L'
646 001366 000167 000140 JMP LOGIC ;YES, RUN 'LOGIC' TEST
647 001372 122767 000103 007672 CMPB #103,INBUF ;TEST FOR 'C'
648 001400 001002 BNE .+6 ;NOT 'C'
649 001402 000167 004556 JMP CALBRT ;YES, RUN 'CALIBRATION' TEST
650 001406 122767 000122 007656 CMPB #122,INBUF ;TEST FOR 'R'
651 001414 001002 BNE .+6 ;NOT 'R'
652 001416 000167 004776 JMP REPTST ;YES, RUN 'REPEATIBILITY' TEST
653 001422 122767 000107 007642 CMPB #107,INBUF ;TEST FOR 'G'
654 001430 001002 BNE .+6 ;NOT 'G'
655 001432 000167 005416 JMP GAIN ;YES, RUN 'GAIN' TEST
656 001436 105767 013436 TSTB SUBX ;TEST FOR SUBTEST
657 001442 001402 BEQ .+6 ;BRANCH IF NOT SET
658 001444 000167 007216 JMP TESTX ;OTHERWISE RUN LOGIC SUBTEST
659 001450 022767 000105 007614 CMP #105,INBUF ;TEST FOR 'E'
660 001456 001002 BNE .+6 ;NOT 'E'
661 001460 000167 007002 JMP RECVRY ;YES, RUN RECOVERY TEST
662 001464 104000 INIT3: PRINT ;ILLEGAL ENTRY
663 001466 014052 QMARK ;TYPE '?'
664 001470 000717 BP INIT2 ;WAIT AGAIN

```

```

665 001472 016701 013366          SETUP:  MOV      ADCS,ADCSR      ;LOAD R1 WITH ADCSR ADDRESS
666 001476 016704 013370          MOV      ADINT,INTVC      ;INITIALIZE INTERRUPT ADDRESS POINTER
667 001502 012767 004000 011704    MOV      #4000,ICOUNT     ;INITIALIZE MAXIMUM ITERATION COUNT
668 001510 012767 001562 011702    MOV      #TEST0+2,RETURN  ;SETUP FIRST RETURN ADDRESS FOR SCOPE
669 001516 012777 000340 013322    MOV      #340,SPSW       ;SET PROCESSOR PRIORITY TO 7
670 001524 005067 013352          CLR      INTFLG          ;INITIALIZE INTERRUPT FLAG
671 001530 000207
672
673
674
675 001532 104000          ;CHECK TO SEE THAT CSR WAS CORRECTLY INITIALIZED
676 001534 014055          LOGIC:  PRINT
677 001536 004767 177730          MESS
678 001542 105777 013306          RESTRT: JSR      %7,SETUP  ;TYPE 'LOGIC' TEST
679 001546 100375          TSTB   #TPS              ;INITIALIZE
680 001550 000005          BPL    .-4               ;WAIT FOR TTY READY
681 001552 005711          RESET  ;REINITIALIZE PERIPHERALS
682 001554 001401          TST   #ADCSR            ;CHECK FOR CORRECT INITIALIZATION OF CSR
683 001556 104400          BEQ   .+4               ;BRANCH IF ALL ZERO
684 001560 104002          ERROR ;CSR WAS NOT INITIALIZED TO ZERO
685
686 001562 005777 013302          TEST0: SCOPE
687 001566 005011          ;SHOW THAT READ/WRITE BITS OF THE CSR CAN BE SET TO 0 OR 1 AND READ BACK
688 001570 032711 177577          TST   #ADDBR            ;CLEAR DONE
689 001574 001401          CLR   #ADCSR            ;SET TO 0
690 001576 104400          BIT   #177577,#ADCSR   ;CHECK ALL BITS BUT DONE
691 001600 005777 013264          BEQ   .+4               ;BRANCH IF ALL BITS READ BACK AS ZERO
692 001604 052777 000340 013234          ERROR ;NOT ALL BITS READ BACK AS ZERO
693 001612 012711 177777          TST   #ADDBR            ;CLEAR DONE
694 001616 011102          BIS   #340,SPSW        ;SET PROCESSOR PRIORITY AT 7 TO PREVENT INTERRUPTS
695 001620 042732 000200          MOV   #177777,#ADCSR  ;LOAD ONES INTO ALL WRITEABLE BITS
696 001624 020227 177536          MOV   #ADCSR,R2        ;DONE MAY OR MAY NOT BE SET
697 001630 001401          BIC   #200,R2          ;SO READ BACK CONTENTS OF THE REGISTER
698 001632 104400          CMP   R2,#177536      ;BUT IGNORE DONE
                          BEQ   .+4               ;BRANCH IF CORRECT
                          ERJ

```

```

699
700
701 001634 104002 TEST1: SCOPE
702 :TEST FOR DONE BEING SET WITHIN 100 MICROSECONDS AFTER A/D START
703 :TEST READING DBR RESETS DONE TO ZERO
704 :TEST INDIRECTLY FOR DONE CLEARING A/D START (VIA SECOND A/D START)
705 001636 105011 CLRB 2ADCSR ;INITIALIZE CSR LOWER BYTE
706 001640 005777 013224 TST 2ADDBR ;READ DBR TO CLEAR DONE
707 001644 105711 TSTB 2ADCSR ;CHECK DONE BIT
708 001646 100001 BPL .+4 ;CONTINUE IF CLEARED
709 001650 104400 ERROR ;READING DBR DIDN'T CLEAR DONE
710 001652 017767 013170 013234 MOV 2PSW,PROC ;SAVE CONDITION CODES
711 001660 105211 INCB 2ADCSR ;START CONVERSION
712 001662 016767 013216 013230 MOV DELAY1,COUNT ;LOAD COUNTER FOR TIMING LOOP
713 001670 005267 013224 INC COUNT ;LOOP FOR ABOUT 100 MICROSECONDS
714 001674 001375 BNE .-4
715 001676 105711 TSTB 2ADCSR ;CHECK DONE BIT
716 001700 100401 BMI .+4 ;CONTINUE IF SET
717 001702 104400 ERROR ;DONE WASN'T SET WITHIN 100 MICROSECONDS.
718 ;EITHER A/D DONE PREVIOUSLY FAILED TO CLEAR A/D START,
719 ;OR DONE ITSELF FAILED HERE
720 001704 016777 013204 013134 MOV PROC,2PSW ;RESTORE CONDITION CODES
721
722 001712 104002 TEST2: SCOPE
723 ;SHOW THAT A CLEAR INSTRUCTION DOESN'T CLEAR DONE
724 001714 005777 013150 TST 2ADDBR ;CLEAR DONE
725 001720 012711 000001 MOV #1,2ADCSR ;INITIALIZE CONVERTER AND START CONVERSION
726 001724 105711 TSTB 2ADCSR ;WAIT FOR DONE
727 001726 100376 BPL .-2
728 001730 105011 CLRB 2ADCSR ;CLEAR LOWER BYTE
729 001732 105711 TSTB 2ADCSR ;CHECK DONE
730 001734 100401 BMI .+4 ;BRANCH IF SET
731 001736 104400 ERROR ;DONE WAS CLEARED
732
733 001740 104002 TEST3: SCOPE
734 ;DATOB TO LOWER BYTE OF CSR SHOULDN'T START CONVERSION
735 001742 005777 013122 TST 2ADDBR ;CLEAR DONE
736 001746 105011 CLRB 2ADCSR ;DATOB TO LOWER BYTE
737 001750 016767 013130 013142 MOV DELAY1,COUNT ;WAIT
738 001756 005267 013136 INC COUNT
739 001762 001375 BNE .-4
740 001764 105711 TSTB 2ADCSR ;CHECK DONE BIT
741 001766 100003 BPL .+10 ;BRANCH IF NOT SET
742 001770 104400 ERROR ;DATOB TO LOWER BYTE CAUSED CONVERSION
743 001772 005777 013072 TST 2ADDBR ;CLEAR DONE
744 001776 105011 CLRB 2ADCSR ;REPEAT TEST, KNOWING BIT1=0
745 002000 016767 013100 013112 MOV DELAY1,COUNT ;WAIT AGAIN
746 002006 005267 013106 INC COUNT
747 002012 001375 BNE .-4
748 002014 105711 TSTB 2ADCSR ;CHECK DONE BIT
749 002016 100001 BPL .+4 ;CONTINUE IF CLEARED
750 002020 104400 ERROR ;DATOB TO LOWER BYTE CAUSED CONVERSION
751

```

```

752 002022 104002 TEST4: SCOPE
753 ;NEW MUX AND GAIN VIA DATO WITH BIT1=0 SHOULD START CONVERSION
754 002024 005777 013040 TST @ADDBR ;CLEAR DONE
755 002030 105011 CLR @ADCSR ;CLEAR BIT1 IN CASE PREVIOUSLY SET
756 002032 005011 CLR @ADCSR ;START CONVERSION
757 002034 016767 013044 013056 MOV DELAY1,COUNT ;WAIT AWHILE
758 002042 005267 013052 INC COUNT
759 002046 001375 BNE -4
760 002050 105711 TSTB @ADCSR ;CHECK DONE BIT
761 002052 100401 BMI .+4 ;CONTINUE IF SET
762 002054 104400 ERROR ;DONE WASN'T SET WITHIN ABOUT 100 MICROSECONDS
763
764 002056 104002 TEST5: SCOPE
765 ;NEW MUX AND GAIN VIA DATOB (BIT1=0) SHOULD START CONVERSION
766 002060 005777 013004 TST @ADDBR ;CLEAR DONE
767 002064 105011 CLR @ADCSR ;CLEAR BIT1
768 002066 105077 012774 CLR @ADCSO ;START CONVERSION VIA DATOB TO UPPER BYTE
769 002072 016767 013006 013020 MOV DELAY1,COUNT ;SET UP TIMING LOOP
770 002100 005267 013014 INC COUNT ;WAIT
771 002104 001375 BNE -4
772 002106 105711 TSTB @ADCSR ;CHECK DONE BIT
773 002110 100401 BMI .+4 ;CONTINUE IF SET
774 002112 104400 ERROR ;DONE WASN'T SET WITHIN ABOUT 100 MICROSECONDS
775
776 002114 104002 TEST6: SCOPE
777 ;NEW MUX AND GAIN WITH BIT 1 PREVIOUSLY SET SHOULDN'T START CONVERSION
778 ;NOTE THAT THE EFFECT OF BIT1 DEPENDS ON ITS STATE BEFORE THE DATO
779 002116 152711 000002 BISR #2,@ADCSR ;SET EXTERNAL ENABLE
780 002122 005777 012742 TST @ADDBR ;CLEAR DONE
781 002126 005011 CLR @ADCSR ;DATO TO CSR
782 002130 016767 012750 012762 MOV DELAY1,COUNT ;WAIT AWHILE
783 002136 005267 012756 INC COUNT
784 002142 001375 BNE -4
785 002144 105711 TSTB @ADCSR ;CHECK DONE BIT
786 002146 100001 BPL .+4 ;CONTINUE IF CLEARED
787 002150 104400 ERROR ;CONVERSION OCCURRED
788
789 002152 104002 TEST7: SCOPE
790 ;SETTING EXTERNAL ENABLE TO ONE VIA FULL WORD DATO SHOULD CAUSE CONVERSION
791 ;IF BIT1 WAS PREVIOUSLY ZERO
792 002154 005777 012710 TST @ADDBR ;CLEAR DONE
793 002160 105011 CLR @ADCSR ;CLEAR LOW BYTE OF STATUS REGISTER
794 002162 012711 000002 MOV #2,@ADCSR ;SET EXTERNAL ENABLE
795 002166 016767 012712 012724 MOV DELAY1,COUNT ;WAIT AWHILE
796 002174 005267 012720 INC COUNT
797 002200 001375 BNE -4
798 002202 142711 000002 BICB #2,@ADCSR ;CLEAR EXTERNAL ENABLE
799 002206 105711 TSTB @ADCSR ;CHECK DONE
800 002210 100401 BMI .+4 ;BRANCH IF SET
801 002212 104400 ERROR ;CONVERSION DID NOT OCCUR
802

```

803	002214	104002			TEST8: SCOPE	
804					:SETTING A/D START WITH BIT 1 PREVIOUSLY SET SHOULD START CONVERSION	
805	002216	005777	012646		TST	ADDBR ;CLEAR DONE
806	002222	112711	000002		MOVB	#2,ADCSR ;SET EXTERNAL ENABLE (BIT 1)
807	002226	005211			INC	ADCSR ;SET A/D START
808	002230	016767	012650	012662	MOV	DELAY1,COUNT ;WAIT
809	002236	005267	012656		INC	COUNT
810	002242	001375			BNE	-4
811	002244	105711			TSTB	ADCSR ;CHECK DONE
812	002246	100401			BMI	+4 ;BRANCH IF SET
813	002250	104400			ERROR	;CONVERSION DID NOT OCCUR
814						
815	002252	104002			TEST9: SCOPE	
816					:TEST FOR NO EXTERNAL CONVERSION (DURING BASIC INSTRUCTION TEST NO EXTERNAL	
817					:SIGNAL SHOULD NORMALLY OCCUR).	
818	002254	105011			CLRB	ADCSR ;INITIALIZE CSR LOW BYTE
819	002256	005777	012606		TST	ADDBR ;CLEAR DONE
820	002262	152711	000002		BISB	#2,ADCSR ;ENABLE EXTERNAL CONVERSIONS
821	002266	005067	012626		CLR	COUNT ;INITIALIZE COUNTER
822	002272	105267	012622		INCB	COUNT ;WAIT AWHILE
823	002276	001375			BNE	-4
824	002300	105711			TSTB	ADCSR ;CHECK DONE
825	002302	100001			BPL	TEST10 ;CONTINUE IF NO CONVERSION OCCURRED
826	002304	104400			ERROR	;AN EXTERNAL TRIGGER OCCURRED
827						
828	002306	104002			TEST10: SCOPE	
829					:TWO SUCCESSIVE DATO'S SHOULD SET THE ERROR FLAG	
830	002310	017767	012532	012576	MOV	PSW,PROC ;STORE CONDITION CODES
831	002316	042777	000020	012522	BIC	#20,PSW ;CLEAR TRACE BIT
832	002324	105011			CLRB	ADCSR ;INITIALIZE CSR LOW BYTE
833	002326	005777	012536		TST	ADDBR ;CLEAR DONE
834	002332	005011			CLR	ADCSR ;FIRST DATO, CLEARS ERROR FLAG
835	002334	005011			CLR	ADCSR ;SECOND DATO, SHOULD SET ERROR FLAG
836	002336	005711			TST	ADCSR ;CHECK ERROR BIT
837	002340	100401			BMI	+4 ;CONTINUE IF ERROR WAS SET
838	002342	104400			ERROR	;ERROR FLAG WASN'T SET
839	002344	016777	012544	012474	MOV	PROC,PSW ;RESTORE CONDITION CODES
840						
841	002352	104002			TEST11: SCOPE	
842					:NEW MUX AND GAIN, FOLLOWED IMMEDIATELY BY SETTING A/D START TO 1	
843					:SHOULD SET ERROR FLAG	
844	002354	017767	012466	012532	MOV	PSW,PROC ;STORE CONDITION CODES
845	002362	042777	000020	012456	BIC	#20,PSW ;CLEAR TRACE BIT
846	002370	105711			TSTB	ADCSR ;CLEAR DONE
847	002372	105011			CLRB	ADCSR ;INITIALIZE LOW BYTE OF CSR
848	002374	005011			CLR	ADCSR ;DATO TO CSR
849	002376	005211			INC	ADCSR ;SET A/D START
850	002400	005711			TST	ADCSR ;CHECK ERROR
851	002402	100401			BMI	+4 ;BRANCH IF SET
852	002404	104400			ERROR	;ERROR WAS NOT SET
853	002406	105711			TSTB	ADCSR ;WAIT FOR DONE
854	002410	100376			BPL	-2
855	002412	016777	012476	012426	MOV	PROC,PSW ;RESTORE CONDITION CODES


```

857 002420 104002 TEST12: SCOPE
858 ;BEGINNING CONVERSION AFTER SEEING DONE SET SHOULD CAUSE AN ERROR IF
859 ;THE PREVIOUS DATA WASN'T READ FROM THE DBR
860 ;SINCE 2 CONSECUTIVE INCB'S WILL NOT CAUSE AN ERROR,
861 ;THE CASE OF AN ERROR DUE TO DATA NOT READ IS DISTINGUISHED
862 ;FROM AN ERROR DUE TO PREVIOUS CONVERSION STILL IN PROCESS
863 002422 017767 012420 012464 MOV @PSW,PROC ;STORE CONDITION CODES
864 002430 042777 000020 012410 BIC #20,@PSW ;CLEAR TRACE BIT
865 002436 005777 012426 TST @ADDBR ;CLEAR DONE
866 002442 012711 000001 MOV #1,@ADCSR ;INITIALIZE CSR AND START CONVERSION
867 002446 105711 TSTB @ADCSR ;WAIT FOR DONE
868 002450 100376 BPL .-2
869 002452 105211 INCB @ADCSR ;START SECOND CONVERSION
870 002454 005711 TST @ADCSR ;CHECK ERROR BIT
871 002456 100401 BMI .+4 ;CONTINUE IF ERROR FLAG SET
872 002460 104400 ERROR ;ERROR FLAG WASN'T SET
873 002462 016777 012426 012356 MOV PROC,@PSW ;RESTORE CONDITION CODES
874
875 002470 104002 TEST13: SCOPE
876 ;CLEARING CSR WITHOUT READING PREVIOUS DATA SHOULD SET ERROR
877 002472 005777 012372 TST @ADDBR ;CLEAR DONE
878 002476 012711 000001 MOV #1,@ADCSR ;INITIALIZE CSR AND CONVERT
879 002502 105711 TSTB @ADCSR ;WAIT FOR DONE
880 002504 100376 BPL .-2
881 002506 005011 CLR @ADCSR ;CLEAR WITHOUT READING DATA
882 002510 005711 TST @ADCSR ;CHECK ERROR BIT
883 002512 100401 BMI .+4 ;CONTINUE IF ERROR FLAG SET
884 002514 104400 ERROR ;ERROR FLAG NOT SET
885 002516 105711 TSTB @ADCSR ;WAIT FOR DONE
886 002520 100376 BPL .-2
887
888 002522 104002 TEST14: SCOPE
889 ;IF DATA IN BUFFER IS READ BEFORE NEXT CONVERSION
890 ;NO ERROR SHOULD OCCUR
891 002524 005777 012340 TST @ADDBR ;CLEAR DONE
892 002530 012711 000001 MOV #1,@ADCSR ;INITIALIZE CSR AND START CONVERSION
893 002534 105711 TSTB @ADCSR ;WAIT FOR DONE
894 002536 100376 BPL .-2
895 002540 005777 012324 TST @ADDBR ;READ DBR
896 002544 105211 INCB @ADCSR ;START SECOND CONVERSION
897 002546 005711 TST @ADCSR ;CHECK ERROR BIT
898 002550 100001 BPL .+4 ;CONTINUE IF NO ERROR
899 002552 104400 ERROR ;ERROR FLAG WAS SET
900 002554 105711 TSTB @ADCSR ;WAIT FOR DONE
901 002556 100376 BPL .-2 ;TO ALLOW LOOPING ON TEST
902

```

903	002560	104002			TEST15: SCOPE	
904					:SETTING A/D START TO ONE TWICE CONSECUTIVELY SHOULD CAUSE	
905					:AN ERROR	
906	002562	017767	012260	012324	MOV @PSW,PROC	;STORE CONDITION CODES
907	002570	042777	000020	012250	BIC #20,@PSW	;CLEAR TRACE BIT
908	002576	005777	012266		TST @ADDBR	;CLEAR DONE
909	002602	012711	000001		MOV #1,@ADCSR	;SET A/D START AND INITIALIZE CSR
910	002606	005211			INC @ADCSR	;SET A/D START AGAIN
911	002610	005711			TST @ADCSR	;CHECK ERROR BIT
912	002612	100401			BMI .+4	;CONTINUE IF NO ERROR
913	002614	104400			ERROR	;ERROR FLAG WAS SET
914	002616	105711			TSTB @ADCSR	;WAIT FOR DONE
915	002620	100376			BPL -2	;TO ALLOW LOOPING ON TEST
916	002622	016777	012266	012216	MOV PROC,@PSW	;RESTORE CONDITION CODES
917						
918	002630	104002			TEST16: SCOPE	
919					:SETTING A/D START, FOLLOWED IMMEDIATELY BY A DATOB TO HIGH BYTE.	
920					:SHOULD SET ERROR FLAG	
921	002632	017767	012210	012254	MOV @PSW,PROC	;STORE CONDITION CODES
922	002640	042777	000020	012200	BIC #20,@PSW	;CLEAR TRACE BIT
923	002646	005777	012216		TST @ADDBR	;CLEAR DONE
924	002652	105011			CLRB @ADCSR	;INITIALIZE LOW BYTE OF CSR
925	002654	105211			INCB @ADCSR	;START CONVERSION
926	002656	005011			CLR @ADCSR	;DATOB TO HIGH BYTE OF CSR
927	002660	005711			TST @ADCSR	;WHICH ALSO CLEARS PREVIOUS ERROR FLAG
928	002662	100401			BMI .+4	;CHECK ERROR FLAG AND BRANCH IF NOT SET
929	002664	104400			ERROR	;ERROR FLAG WAS SET
930	002666	105711			TSTB @ADCSR	;WAIT FOR DONE TO
931	002670	100376			BPL -2	;ALLOW LOOPING ON TEST
932	002672	016777	012216	012146	MOV PROC,@PSW	;RESTORE CONDITION CODES
933						
934	002700	104002			TEST17: SCOPE	
935					:DATO WITH BIT 1=1 SHOULDN'T CAUSE ERROR EVEN THOUGH	
936					:PREVIOUS DATA WAS NOT READ	
937	002702	005777	012162		TST @ADDBR	;CLEAR DONE
938	002706	105011			CLRB @ADCSR	;INITIALIZE CSR LOW BYTE
939	002710	012711	000002		MOV #2,@ADCSR	;CLEAR ERROR,SET EXTERNAL ENABLE.
940	002714	105711			TSTB @ADCSR	;AND INITIATE CONVERSION
941	002716	100376			BPL -2	;WAIT FOR DONE
942	002720	005011			CLR @ADCSR	;SECOND DATO, WITHOUT READING DATA
943	002722	005711			TST @ADCSR	;CHECK ERROR BIT
944	002724	100001			BPL .+4	;BRANCH IF NOT SET
945	002726	104400			ERRJP	;ERROR WAS SET
946						

947	002730	104002			TEST18: SCOPE	
948					;DONE SHOULD CAUSE AN INTERRUPT IF INTERRUPT ENABLE IS SET	
949	002732	005777	012132		TST @ADDBR	;CLEAR DONE
950	002736	105011			CLRB @ADCSR	;INITIALIZE LOW BYTE OF CSR
951	002740	052777	000340	012100	BIS #340,@PSW	;SET PROCESSOR PRIORITY TO LEVEL 7
952	002746	012714	003014		MOV #TINTD,@INTVC	;SETUP RETURN FROM INTERRUPT
953	002752	017764	012070	000002	MOV @PSW, 2(INTVC)	
954	002760	042777	000340	012060	BIC #340,@PSW	;SET PROCESSOR PRIORITY TO 0
955	002766	012711	000001		MOV #1,@ADCSR	;CLEAR ERROR AND START CONVERSION
956	002772	112711	000100		MOVB #100,@ADCSR	;SET INTERRUPT ENABLE
957	002776	105711			TSTB @ADCSR	;CHECK DONE
958	003000	100376			BPL .-2	
959	003002	104400			ERROR	;DONE DIDN'T CAUSE AN INTERRUPT
960	003004	016477	000002	012034	MOV 2(INTVC),@PSW	;RESTORE PROCESSOR STATUS
961	003012	000401			BR .+4	;SKIP NEXT INSTRUCTION
962	003014	022626			TINTD: CMP (6)+,(6)+	;RESTORE SP
963	003016	105011			CLRB @ADCSR	;CLEAR INTERRUPT ENABLE
964	003020	012714	000132		MOV #132,@INTVC	;CHANGE INTERRUPT POINTER TO
965	003024	005037	000132		CLR @#132	;CAUSE HALT
966						
967	003030	104002			TEST19: SCOPE	
968					;ERROR SHOULD CAUSE AN INTERRUPT IF INTERRUPT ENABLE IS SET	
969	003032	012714	003106		MOV #TINTE,@INTVC	;SETUP INTERRUPT RETURN
970	003036	052777	000340	012002	BIS #340,@PSW	
971	003044	017764	011776	000002	MOV @PSW, 2(INTVC)	
972	003052	105011			CLRB @ADCSR	;PREVENT INTERRUPT FROM OCCURRING
973	003054	042777	000340	011764	BIC #340,@PSW	;SET PROCESSOR TO LOWEST PRIORITY
974	003062	005777	012002		TST @ADDBR	;CLEAR DONE
975	003066	012711	100100		MOV #100100,@ADCSR	;START CONVERSION, ENABLE INTERRUPTS, SET ERROR
976	003072	000240			NOP	;CLOCK IN INTERRUPT
977	003074	052777	000340	011744	BIS #340,@PSW	;RAISE PROCESSOR PRIORITY
978	003102	104400			ERROR	;NO INTERRUPT OCCURRED
979	003104	000401			BR .+4	
980	003106	022626			TINTE: CMP (6)+,(6)+	;RESTORE SP
981	003110	105011			CLRB @ADCSR	;CLEAR INTERRUPT ENABLE
982	003112	012714	000132		MOV #132,@INTVC	;CHANGE INTERRUPT RETURN
983	003116	005037	000132		CLR @#132	;TO CAUSE A HALT
984	003122	105711			TSTB @ADCSR	;WAIT FOR DONE
985	003124	100376			BPL .-2	
986						

```

987 003126 104002 TEST20: SCOPE
988 ;NO INTERRUPT SHOULD OCCUR IF PROCESSOR IS AT LEVEL 7 (BIT 2=0)
989 003130 052777 000340 011710 BIS #340, @PSW ;SET PROCESSOR PRIORITY TO LEVEL 7
990 003136 012714 003164 MOV #TINT7A, @INTVC ;SETUP RETURN
991 003142 017764 011700 000002 MOV @PSW, 2(INTVC)
992 003150 105011 CLRB @ADCSR ;MAKE SURE THAT CONVERSION WILL OCCUR
993 003152 012711 100100 MOV #100100, @ADCSR ;ENABLE INTERRUPTS, SET ERROR, AND START CONVERSION
994 003156 000240 NOP ;CLOCK INTERRUPT
995 003160 105011 CLRB @ADCSR ;DISABLE INTERRUPTS
996 003162 000402 BR .+6 ;CONTINUE IF NO INTERRUPT
997 003164 104400 TINT7A: ERROR ;INTERRUPT OCCURRED WITH PROCESSOR AT LEVEL 7
998 003166 022626 CMP (6)+, (6)+ ;RESTORE STACK POINTER
999 003170 105711 TSTB @ADCSR ;WAIT FOR DONE
1000 003172 100376 BPL .-2
1001 003174 012714 000132 MOV #132, @INTVC ;CHANGE INTERRUPT RETURN
1002 003200 005037 000132 CLR @#132 ;TO CAUSE A HALT

```

1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

```

.MACR INT STATUS, N, TESTNO
:AN INTERRUPT SHOULD BE SEEN IF PROCESSOR IS AT LEVEL 'N' (BIT 2=0)
TESTNO: SCOPE
BIS #340, @PSW ;SET PROCESSOR PRIORITY TO LEVEL 7
MOV #TINT'N'A, @INTVC ;SETUP RETURN
MOV @PSW, 2(INTVC)
MOV #1, @ADCSR ;INITIALIZE CSR
TSTB @ADCSR ;WAIT FOR DONE
BPL .-2
MOV #100, @ADCSR ;ENABLE INTERRUPTS
MOV #STATUS, @PSW ;SET PROCESSOR PRIORITY TO LEVEL 'N'
MOV 2(INTVC), @PSW ;SET PROCESSOR BACK TO LEVEL 7
ERROR ;NO INTERRUPT OCCURRED
BR .+4
TINT'N'A: POP2SP ;RESTORE STACK POINTER
CLRB @ADCSR ;CLEAR INTERRUPT ENABLE
MOV #132, @INTVC ;CHANGE INTERRUPT RETURN ADDRESS
CLR @#132 ;TO CAUSE A HALT
.ENDM

```

1026
1027
1028
1029
1030
1031
1032

DC3204
DC3276

INT 300.6,TEST21
INT 240.5,TEST22

C03

AD02 AC11 DIAGNOSTIC
DZADAC.P11

MACY11 27(732) 16-SEP-76 16:37 PAGE 28

1333 003372
1334
1335 003462
1336

INT 200.4,TEST23

INT 140.3,TEST24

D03

AD02/AD11 DIAGNOSTIC
DZADAC.P11

MACY11 27(732) 16-SEP-76 16:37 PAGE 29

1037 003554
1038
1039 003646
1040

INT 100.2,TEST25

INT 40.1,TEST26

E03

RO02/RO11 DIAGNOSTIC
DZADAC.P11

MACY11 27(732) 16-SEP-76 16:37 PAGE 30

```

1041 :FIND THE LEVEL AT WHICH BIT 2=1 CAUSES AN INTERRUPT
1042 :PRINT OUT A MESSAGE TELLING WHERE THE INTERRUPT OCCURRED
1043 :MAKE SURE THAT IT ALWAYS OCCURS AT THE SAME LEVEL
1044 :FIRST SEE THAT NO INTERRUPT OCCURS WITH PROCESSOR
1045 :A* LEVEL 7 PRIORITY
1046 003740 104002 TEST27: SCOPE ;NO INTERRUPT SHOULD OCCUR WITH PROCESSOR AT LEVEL 7
1047 003742 052777 000340 011076 BIS #340, @PSW ;SET PROCESSOR TO LEVEL 7
1048 003750 005777 011114 TST @ADDBR ;CLEAR DONE
1049 003754 105011 CLRB @ADCSR ;INITIALIZE LOW BYTE OF STATUS REGISTER
1050 003756 005011 CLR @ADCSR ;INITIALIZE STATUS REGISTER
1051 003760 105711 TSTB @ADCSR ;WAIT FOR DONE
1052 003762 100376 BPL -2
1053 003764 012714 004006 MOV #TINT7B, @INTVC ;SETUP RETURN
1054 003770 017764 011052 000002 MOV @PSW, 2(@INTVC) ;SETUP RETURN PROCESSOR STATUS
1055 003776 112711 000104 MOVB #104, @ADCSR ;SET INTERRUPT ENABLE AND BIT 2
1056 004002 000240 NOP ;CLOCK INTERRUPT
1057 004004 000402 BR +6 ;BRANCH IF NO INTERRUPT
1058 004006 104400 *INT7B: ERROR ;INTERRUPT OCCURRED WITH PROCESSOR AT LEVEL 7
1059 004010 022626 CMP (SP)+, (SP)+ ;RESTORE SP
1060 004012 105011 CLRB @ADCSR ;DISABLE INTERRUPTS
1061 004014 012714 000132 MOV #132, @INTVC ;CHANGE INTERRUPT RETURN
1062 004020 005037 000132 CLR @#132 ;TO CAUSE A HALT
1063
1064 :MAKE SURE AN INTERRUPT CAN OCCUR WITH PROCESSOR PRIORITY AT LEVEL 0
1065 004024 104002 *EST28: SCOPE
1066 004026 052777 000340 011012 BIS #340, @PSW ;SET PROCESSOR TO LEVEL 7
1067 004034 005777 011030 TST @ADDBR ;CLEAR DONE
1068 004040 105011 CLRB @ADCSR ;CLEAR EXTERNAL ENABLE
1069 004042 005011 CLR @ADCSR ;INITIALIZE STATUS REGISTER
1070 004044 105711 TSTB @ADCSR ;WAIT FOR DONE
1071 004046 100376 BPL -2
1072 004050 012714 004112 MOV #TINTX, @INTVC ;SET UP RETURN
1073 004054 017764 010756 000002 MOV @PSW, 2(@INTVC) ;SETUP RETURN PROCESSOR STATUS
1074 004062 042777 000340 010756 BIC #340, @PSW ;SET PROCESSOR TO LEVEL 0
1075 004070 112711 000104 MOVB #104, @ADCSR ;SET INTERRUPT ENABLE AND BIT 2
1076 004074 000240 NOP ;CLOCK IN INTERRUPT
1077 004076 105011 CLRB @ADCSR ;DISABLE INTERRUPTS
1078 004100 052777 000340 010740 BIS #340, @PSW ;RESTORE PROCESSOR PRIORITY TO 7
1079 004106 104400 ERROR ;NO INTERRUPT OCCURRED
1080 004110 000402 BR +6 ;SKIP NEXT 2 INSTRUCTIONS
1081 004112 105011 TINTX: CLRB @ADCSR ;CLEAR INTERRUPT ENABLE AFTER INTERRUPT
1082 004114 022626 CMP (SP)+, (SP)+ ;RESTORE STACK POINTER
1083 004116 012714 000132 MOV #132, @INTVC ;CHANGE INTERRUPT RETURN
1084 004122 005037 000132 CLR @#132 ;TO CAUSE A HALT

```


1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123

```

.MACR INT2 TSTNO,PRP,STTS,INTLVL,NXTST
:TEST FOR AN INTERRUPT ON LEVEL 'INTLVL' WITH BIT2 SET
TSTNO: SCOPE
      BIS #340, @PSW ;SET PROCESSOR PRIORITY TO 7
      TST @ADDBR ;CLEAR DOI
      CLRB @ADCSR ;CLEAR EXTERNAL ENABLE
      CLR @ADCSR ;INITIALIZE STATUS REGISTER
      TSTB @ADCSR ;WAIT FOR DONE
      BPL .-2
      MOV @TINT'PRP'B, @INTVC ;SETUP RETURN
      MOV @PSW, 2(@INTVC) ;SETUP RETURN PROCESSOR STATUS
      MOV @STTS, @PSW ;SET PROCESSOR TO LEVEL 'PRP' PRIORITY
      MOVB #104, @ADCSR ;SET INTERRUPT ENABLE AND BIT 2
      NOP ;CLOCK INTERRUPT
      CLRB @ADCSR ;DISABLE INTERRUPTS
      MOV 2(@INTVC), @PSW ;RESTORE PROCESSOR STATUS
      MOV #132, @INTVC ;CHANGE INTERRUPT RETURN
      CLR @132 ;TO CAUSE A HALT
      BR NXTST ;GO TO NEXT TEST

TINT'PRP'B: CLRB @ADCSR ;DISABLE FURTHER INTERRUPTS
            MOV #132, @INTVC ;CHANGE INTERRUPT RETURN
            CLR @132 ;TO CAUSE A HALT
            CMP (SP)+, (SP)+ ;RESTORE STACK POINTER
            TST INTFLG ;TEST TO INHIBIT TYPE OUT
            BNE SET'INTLVL ;BRANCH IF FLAG SET
            MOV #INTLVL, INTFLG ;SET FLAG AND LEVEL
            PRINT ;SETUP FOR PRINTOUT
            MESG ;PRINT MESSAGE 'BIT2 SET CAUSES AN
            PRTOCT ;PRINT LEVEL NUMBER
            INTFLG
            PRINT ;CARRIAGE RETURN, LINEFEED
            CRLF

SET'INTLVL: CMP INTFLG, #INTLVL ;CHECK PREVIOUS LEVEL
            BPL NXTST ;GO TO NEXT TEST IF INTERRUPT PREVIOUSLY AT AN EQUAL
            ;OR HIGHER LEVEL
            ERROR ;INTERRUPT OCCURRED PREVIOUSLY ONLY AT A LOWER LEVEL
            .ENDM

```

```

1124
1125 004126          INT2    TEST29,6,300,7,TEST30
1126
1127 004300          INT2    TEST30,5,240,6,TEST31
1128
1129 004452          INT2    TEST31,4,200,5,TEST32
1130
1131 004624          INT2    TEST32,3,140,4,TEST33
1132
1133 004776          INT2    TEST33,2,100,3,TEST34
1134
1135 005150          INT2    TEST34,1,40,2,TEST35
1136
1137 005322          INT2    TEST35,0,0,1,TEST36
1138          :NO INTERRUPT SHOULD OCCUR IF INTERRUPT ENABLE IS SET AND ERROR AND DONE ARE
1139          :BOTH CLEAR
1140          TEST36: SCOPE
1141 005474 104002          TST     @ADDBR          ;CLEAR DONE TO PREVENT ERROR
1142 005476 005777 007366  CLRB   @ADCSR          ;CLEAR EXTERNAL ENABLE
1143 005502 105011          CLR     @ADCSR          ;INITIALIZE STATUS REGISTER
1144 005504 005011          TSTB   @ADCSR          ;WAIT FOR DONE
1145 005506 105711          BPL    -2
1146 005510 100376          TST     @ADDBR          ;CLEAR DONE
1147 005512 005777 007352  MOV     @NINT,@INTVC   ;SETUP RETURN IN CASE
1148 005516 C12714 005562          BIS    @340,@PSW
1149 005522 052777 000340 007316  MOV     @PSW,2(INTVC)
1150 005530 017764 007312 000002  BIS    @340,@PSW
1151 005536 042777 000340 007302  BIC    @340,@PSW          ;SET PROCESSOR TO LOWEST PRIORITY
1152 005544 112711 000100          MOVB   @100,@ADCSR     ;ENABLE INTERRUPTS
1153 005550 000240          NOP
1154 005552 052777 000340 007266  BIS    @340,@PSW
1155 005560 000402          BR     .+6
1156 005562 104400          NINT:  ERROR
1157 005564 022626          CMP    (SP)+,(SP)+
1158 005566 012714 000132  MOV     @132,@INTVC
1159 005572 005037 000132  CLR    @132
1160 005576 105011          CLRB   @ADCSR          ;DISABLE INTERRUPTS
1161
1162          :CHECK FOR REPEATABILITY OF READING DBR AT EACH GAIN
1163          TEST37: SCOPE
1164 005600 104002          TST     @ADDBR          ;CLEAR DONE
1165 005602 005777 007262  CLRB   @ADCSR          ;CLEAR LOW BYTE
1166 005606 105011          CLR     @ADWRD2        ;LOAD INITIAL CHANNEL
1167 005610 005067 007260  JSR    %7,CHKGAN       ;CHECK REPEATABILITY
1168 005614 004767 000034          BIS    @10,@ADWRD2    ;SET GAIN TO X2
1169 005620 052767 000010 007246  MOVB   @20,@ADWRD2    ;SET GAIN TO X4
1170 005626 112767 000020 007240  JSR    %7,CHKGAN       ;CHECK REPEATABILITY
1171 005634 004767 000014          BIS    @30,@ADWRD2    ;SET GAIN TO X8
1172 005640 042767 000030 007226  JSR    %7,CHKGAN       ;CHECK REPEATABILITY
1173 005646 004767 000002          BR     TEST38
1174 005652 000421          CHKGAN: MOV     @100,COUNT     ;SET UP COUNTER
1175 005654 012767 000100 007236  MOV     @ADWRD2,@ADCSR ;CONVERT AT DESIRED GAIN
1176 005662 016711 007206  TSTB   @ADCSR          ;WAIT FOR DONE
1177 005666 105711          BPL    -2
1178 005670 100376          MOV     @ADDBR,R0
1179 005672 017700 007172  CMP    @ADDBR,R0
1180 005676 027700 007166  BEQ    .+4
1181 005702 001401          ;LOAD DATA INTO REGISTER 0
1182          ;REPEAT DATA AND COMPARE
1183          ;CONTINUE IF BOTH ARE THE SAME

```

1180	005704	104400			ERROR			; 2 READINGS OF DBR WEREN'T THE SAME
1181	005706	005367	007206		DEC	COUNT		; COUNT DOWN
1182	005712	001371			BNE	RELOOP		; LOOP
1183	005714	000207			RTS	%7		; RETURN
1194	005716	104002						
1185					TEST38: SCOPE			
1186					; INIT PULSE SHOULD INITIALIZE THE INTERNAL "READ" FLIP-FLOP OFF			
1187	005720	012767	000002	005466	; IF THIS FAILS, THE VERY FIRST CONVERSION WILL SET ERROR			
1188	005726	105777	007122		MOV	#2, ICOUNT		; RUN TEST TWICE ONLY
1189	005732	100375			TSTB	BTFS		; WAIT FOR TTY READY TO AVOID
1190	005734	000005			BPL	.-4		; CLOBBERING ANY OUTPUT
1191	005736	005211			RESET			; SEND OUT INIT PULSE
1192	005740	005711			INC	@ADCSR		; START CONVERSION
1193	005742	100001			TST	@ADCSR		; CHECK ERROR
1194	005744	104400			BPL	.+4		; BRANCH IF NOT SET
1195	005746	105711			ERROR			; INTERNAL FLIP-FLOP INCORRECTLY INITIALIZED
1196	005750	100375			TSTB	@ADCSR		; WAIT FOR DONE
1197					BPL	.-2		

```

1198 005752 104002 TEST39: SCOPE
1199 ;MAKE SURE THAT WITH BIT 0=0, INTERRUPTS DON'T OCCUR AT BOTH LEVELS AT ONCE
1200 005754 012767 004000 005432 †STDN: MOV #4000,ICOUNT
1201 005762 105777 007102 TSTB @ADDBR ;CLEAR DONE
1202 005766 105011 CLRB @ADCSR ;INITIALIZE LOW BYTE OF CSR
1203 005770 052777 000340 007050 BIS #340,@PSW ;SET PROCESSOR TO LEVEL 7
1204 005776 012714 006044 MOV #TINT2,@INTVC ;SETUP RETURN FROM INTERRUPT
1205 006002 017764 007040 000002 MOV @PSW,2(@INTVC)
1206 006010 042777 000340 007030 BIC #340,@PSW ;SET PROCESSOR PRIORITY TO LEVEL 0
1207 006016 012711 000001 MOV #1,@ADCSR ;CLEAR ERROR AND START CONVERSION
1208 006022 112711 000100 MOVB #100,@ADCSR ;SET INTERRUPT ENABLE
1209 006026 105711 TSTB @ADCSR ;CHECK DONE
1210 006030 100376 BPL .-2 ;BRANCH TO RESTORE ORDER
1211 006032 104400 ERROR ;DONE DIDN'T CAUSE AN INTERRUPT
1212 006034 016477 000002 007004 MOV 2(@INTVC),@PSW ;RESTORE PROCESSOR STATUS
1213 006042 000414 BR TINT3+4
1214 006044 022626 TINT2: CMP (6)+,(6)+ ;RESTORE STACK POINTER
1215 006046 012714 006070 MOV #TINT3,@INTVC ;SET UP RETURN FOR ERROR INTERRUPT
1216 006052 005077 006770 CLR @PSW ;SET PROCESSOR TO LEVEL 0
1217 006056 000240 NOP ;ALLOW INTERRUPT
1218 006060 016477 000002 006760 MOV 2(@INTVC),@PSW ;RESTORE PROCESSOR PRIORITY
1219 006066 000402 BR TINT3+4 ;BRANCH AROUND HALT
1220 006070 104400 TINT3: ERROR
1221 006072 022626 CMP (6)+,(6)+ ;RESTORE STACK POINTER
1222 006074 105011 CLRB @ADCSR ;CLEAR INTERRUPT ENABLE
1223 006076 012714 000132 MOV #132,@INTVC ;CHANGE INTERRUPT POINTER TO
1224 006102 005037 000132 CLR @#132 ;CAUSE A HALT
1225 006106 104002 SCOPE
1226
1227 006110 105777 006740 TSTB @TPS
1228 006114 100375 BFL .-4
1229 006116 012777 000207 006732 MOV #207,@TPB
1230 006124 012767 001562 005266 MOV #TEST0+2,RETURN
1231 006132 012767 004000 005254 MOV #4000,ICOUNT
1232 006140 013701 000042 MOV @#42,R1
1233 006144 001405 BEQ LPTEST ;IF NOT RUN UNDER 'DDP1'
1234 006146 000005 RESET ;OTHERWISE EXIT TO MONITOR
1235 006150 004711 JSR PC,(R1)
1236 006152 000240 NOP
1237 006154 000240 NOP
1238 006156 000240 NOP
1239 006160 000167 173352 LPTEST: JMP RESTR ;START INSTRUCTION TESTS OVER
  
```

```

1240 :*****
1241 :CALIBRATION ROUTINE
1242 :*****
1243
1244 :ROUTINE REQUESTS THE TYPE OF 'SYNC' TO BE USED ('I' INTERNAL OR 'E' EXTERNAL).
1245 :THE PROGRAM THEN TAKES CONTINUOUS CONVERSIONS USING DATA SW'S 9-15
1246 :TO SELECT THE CH. SW'S 3&4 TO SELECT GAIN AND EITHER SW'S 0-2 TO SELECT DELAY
1247 :OR SW '7' TO PRINT THE CONVERSION VALUE.
1248
1249 006164 012767 006204 006716 CALBRT: MOV #CALBT1,AVECTR ;SET UP 'A' RESTART ADDRESS
1250 006172 104000 PRINT
1251 006174 014123 MES7 ;TYPE TEST HEADER
1252 006176 104000 PRINT
1253 006200 014177 MES10 ;TEXT 'SYNC I OR E'
1254 006202 000402 BR CALB1A ;WAIT FOR INPUT
1255 006204 104000 CALBT1: PRINT
1256 006206 015036 MES39 ;TEST 'SYNC?'
1257 006210 104011 CALB1A: TTYIN ;WAIT FOR INPUT.
1258 006212 016767 003054 006674 MOV INBUF,PROC ;SAVE IT IN TEMP STORAGE
1259 006220 012767 000001 006672 MOV #1,COUNT ;SET UP FOR 'I' CONVERSION
1260 006226 104000 PRINT
1261 006230 014045 CR LF
1262 006232 012767 000001 005154 MOV #1,ICOUNT ;SETUP TO PRINT 'I' VALUE
1263 006240 117767 006616 006627 CALBT2: MOVB #SWR0,ADWRD2+1 ;GET CH. FROM THE SW REG.
1264 006246 000241 CLC ;CLEAR THE 'C' BIT
1265 006250 106067 006621 RORB ADWRD2+1 ;MOVE WORD RIGHT '1'
1266 006254 117767 006600 006612 MOVB #SWR,ADWRD2 ;GET GAIN FROM SW REG.
1267 006262 142767 000347 006604 BICB #47,ADWRD2 ;CLR UNWANTED BITS, A/D WORD COMPLETE
1268 006270 017767 006564 006624 MOV #SWR,KSTOR1 ;GET DELAY FROM SW REG.
1269 006276 042767 177770 006616 BIC #177770,KSTOR1 ;CLR UNWANTED BITS
1270 006304 017767 006550 006612 MOV #SWR,KSTOR2 ;SAVE ORIGINAL SWITCH SETTING.
1271 006312 022767 000105 006574 CMP #105,PROC ;TEST SYNC SELECT
1272 006320 001003 BNE CALB2A ;BRANCH IF NOT 'E'
1273 006222 052767 000002 006544 BIS #2,ADWRD2 ;OTHERWISE ADD 'EXT' SYNC BIT TO A/D.
1274 006330 004767 003112 CALB2A: JSR PC,ADCNVT ;TAKE AND STORE THE CONVERSIONS
1275 006334 105777 006520 TSTB #SWR ;TEST FOR SW7 TO PRINT
1276 006340 100007 BPL CALB2B ;BRANCH IF NOT SET.
1277 006342 012767 015400 001574 MOV #ADBUFF,AVGTAB ;SET UP TO PRINT VALUE
1278 006350 104014 PRTAVG ;PRINT IT
1279 006352 104000 PRINT
1280 006354 014045 CR LF
1281 006356 000412 BR CALBT4 ;TEST FOR LOOP
1282 006360 016700 007014 CALB2B: MOV ADEJFF,RO ;SET UP A/D BUFFER.
1283 006364 016767 006532 006542 MOV KSTOR1,TEMP2 ;SET UP DELAY (RESET COUNT)
1284 006372 104016 CALBT3: TSTKS ;TEST FOR TTY FLAG
1285 006374 000005 RESET
1286 006376 005367 006532 DEC TEMP2 ;DECREMENT DELAY
1287 006402 100373 BPL CALBT3
1288 006404 104016 CALBT4: TSTKS ;TEST FOR KEYBOARD INTERRUPT
1289 006406 026777 006512 006444 CMP KSTOR2,#SWR ;TEST IF SWITCH REGISTER HAS CHANGED
1290 006414 001745 BEQ CALB2A ;BRANCH AND TAKE NEXT CONVERSION
1291 006416 000710 BR CALBT2 ;YES, COMPUTE NEW INPUT

```

1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342

006420 012767 006432 006462
006426 104000
006430 014250
006432 005067 006502
006436 104000
006440 014276
006442 104011
006444 104001
006446 016767 002764 006446
006454 016767 006442 006442
006462 005767 002752
006466 001407
006470 026767 002744 006424
006476 100755
006500 016767 002734 006416
006506 104003
006510 104000
006512 014311
006514 104011
006516 104001
006520 016767 002712 006400
006526 016767 006370 006374
006534 104016
006536 012767 001000 006354
006544 116767 006360 006323
006552 004767 002670
006556 104004
006560 104005
006562 032777 002000 006270
006570 001047
006572 032777 020000 006260
006600 001116
006602 022767 000004 006316
006610 001005
006612 022767 001000 006416
006620 001033
006622 000505

```
*****  
:REPEATIBILITY TEST  
*****  
:THIS ROUTINE TO DESIGNED TO SHOW REPEATIBILITY BY TAKING A SERIES OF  
: '512' CONVERSIONS AT A SPECIFIED GAIN, AVERAGING THEM AND THEN CATORIZING  
: THEM IN BINS FROM THE AVERAGE PLUS & MINUS 6 COUNTS. THE ROUTINE  
: REQUESTS FOR A CHANNEL OR CHANNELS, A GAIN AND A COUNT SPREAD TO BE TYPED  
: IN VIA THE OPERATOR. A CONTINUOUS SERIES OF CONVERSIONS ARE THEN TAKEN  
: AND COMPARED AGAINST THE INPUT COUNT SPREAD. IF ALL '512' CONVERSIONS  
: ARE FOUND TO BE WITHIN THE SPREAD THE NEXT CH. IS EXERCISED OTHERWISE  
: THE COUNTS ARE TYPED OUT. SETTING SWITCH '10' TO A '1' WILL FORCE A PRINTOUT  
: OF THE CH (S).  
REPTST: MOV #REPT1,AVECTR ;SET UP CNTR 'A' VECTOR ADDRESS  
PRINT  
MES13 ;TEXT 'REPEATIBILITY TEST'  
REPT1: CLR MESPRT  
PRINT  
MES14 ;REQUEST CHANNEL (S)  
TTYIN ;WAIT FOR INPUT  
DECOCT ;CONVERT TO OCTAL  
MOV BCDTAB,KSTOR1 ;SAVE AS INTIAL CH.  
MOV KSTOR1,KSTOR2 ;ALSO SAVE AS 2ND CH. ENTRY  
TST BCDTAB+2 ;TEST FOR SECOND ENTRY  
BEQ REPT2 ;BRANCH IF NO SECOND ENTRY  
CMP BCDTAB+2,KSTOR1 ;COMPARE ENTRY 1 TO ENTRY 2  
BMI REPT1 ;BRANCH AND RESTART IF ILLEGAL  
MOV BCDTAB+2,KSTOR2 ;OTHERWISE SAVE AS SECOND CH.  
REPT2: GAININ ;SET UP GAIN  
PRINT  
MES16 ;TEXT 'COUNT SPREAD ?'  
TTYIN ;WAIT FOR ENTRY  
DECOCT ;DECODE TO OCTAL  
MOV BCDTAB,KSTOR3 ;SAVE IT  
REPT2A: MOV KSTOR1,KSTOR4 ;SAVE STARTING CH.  
REPT3: TSTTKS ;TEST FOR KEYBOARD FLAG  
MOV #1000,COUNT ;SET FOR '512' CONVERSIONS  
MOVB KSTOR4,ADWORD2+1 ;MOV SELECTED CH. TO HIGH BYTE OF ADWORD  
JSR PC,ADCVT ;TAKE THE CONVERSIONS  
CMPUTE ;AVERAGE & COMPUTE DISTRIBUTION  
CATORIZ  
BIT #SW10,ASWR ;TEST DATA SW10  
BNE REPT4 ;IF SET, FORCE TYPE OUT  
TSTCT4: BIT #SW13,ASWR ;TEST FOR INHIBIT TYPEOUT  
BNE REPT7 ;BRANCH IF SW SET  
CMP #4,KSTOR3 ;WAS 4 TYPED  
BNE TSTCT3 ;NO, TEST FOR '3'  
CMP #1000,XSPRD4 ;TOTAL COUNTS WITHIN 4 COUNTS  
BNE REPT4 ;BRANCH IF NO.  
BR REPT7 ;YES, TEST NEXT CH.
```

1343	006624	022767	000003	006274	TSTCT3:	CMP	#3,KSTOR3	;COUNT = TO 3
1344	006632	001005				BNE	TSTCT2	;NO TEST COUNT 2
1345	006634	022767	001000	006372		CMP	#1000,XSPRD3	
1346	006642	001022				BNE	REPT4	;BRANCH IF COUNT NOT WITHIN 3
1347	006644	000474				BR	REPT7	;YES, TEST NEXT CH.
1348								
1349	006646	022767	000002	006252	TSTCT2:	CMP	#2,KSTOR3	;COUNT =TO 2
1350	006654	001005				BNE	TSTCT1	;NO, TEST COUNT 1
1351	006656	022767	001000	006346		CMP	#1000,XSPRD2	
1352	006664	001011				BNE	REPT4	;BRANCH IF NOT WITHIN 2
1353	006666	000463				BR	REPT7	;YES, TEST NEXT CH.
1354	006670	022767	000001	006230	TSTCT1:	CMP	#1,KSTOR3	;COUNT = TO 1
1355	006676	001004				BNE	REPT4	;NO, REPORT EVEN IF NOT '0'
1356	006700	022767	001000	006322		CMP	#1000,XSPRD1	
1357	006706	001453				BEQ	REPT7	;BRANCH IF TOTAL WITHIN 1 COUNT
1358	006710	104000			REPT4:	PRINT		
1359	006712	014045				CRLF		
1360	006714	005767	006220			TST	MESPRT	;TEST IF HEADER HAS BEEN TYPED
1361	006720	001002				BNE	REPT5	;BRANCH IF YES
1362	006722	104000				PRINT		
1363	006724	014342				MES19		;TEXT 'CH. HIGH AVG. LOW'
1364	006726	104016			REPT5:	TSTTKS		;TEST FOR KEYBOARD INTERRUPT
1365	006730	104000				PRINT		
1366	006732	014045				CRLF		;CARRIAGE RETURN, LINE FEED
1367	006734	016702	006170			MOV	KSTOR4,R2	;MOV. CH.
1368	006740	104006				BINDEC		;CONVERT TO DECIMAL AND PRINT
1369	006742	104007				SPACE		
1370	006744	104010				PRTCT		;PRINT LOW VALUE
1371	006746	015146				LOW		
1372	006750	104007				SPACE		
1373	006752	104010				PRTCT		;PRINT AVERAGE VALUE
1374	006754	015162				AVRAGE		
1375	006756	104007				SPACE		
1376	006760	104010				PRTCT		;PRINT HIGH VALUE
1377	006762	015144				HIGH		
1378	006764	005767	006150			TST	MESPRT	
1379	006770	001002				BNE	REPT6	
1380	006772	104000				PRINT		
1381	006774	014374				MES20		;PRINT 'COUNT SPREAD' HEADER
1382	006776	022767	000007	006134	REPT6:	BIS	#7,MESPRT	;INHIBIT OTHER HEADERS
1383	007004	022767	001000	006200		CMP	#1000,AVGCNT	;TEST IF ALL COUNTS WERE AT AVG.
1384	007012	001411				BEQ	REPT7	;BRANCH TO NEXT CH. IF YES.
1385	007014	104000				PRINT		
1386	007016	014045				CRLF		
1387	007020	012704	015176			MOV	#ORLOW,R4	
1388	007024	012402			REPT6A:	MOV	(R4)+,R2	
1389	007026	104006				BINDEC		;TYPE OUT COUNT SPREAD
1390	007030	022704	015230			CMP	#XSPRD1,R4	;TEST FOR DONE
1391	007034	001373				BNE	REPT6A	;BRANCH IF NO AND TYPE NEXT COUNT
1392	007036	005267	006066		REPT7:	INC	KSTOR4	;INCREMENT 'CH.'
1393	007042	026767	006056	006060		CMP	KSTOR2,KSTOR4	;TESTED ALL CH.(S)?
1394	007050	002231				BGE	REPT3	;BRANCH IF NO AND TEST NEXT CH.
1395	007052	000625				BR	REPT2A	;OTHERWISE RESET AND REPEAT

1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446

007054 012767 007066 006026
007062 104000
007064 013522
007066 104000
007070 014502
007072 104011
007074 104001
007076 116767 002334 005771
007104 005067 006030

007110 104000
007112 014512
007114 104012
007116 000000
007120 015240
007122 015264
007124 005767 005746
007130 001406

007132 104000
007134 014521
007136 104012
007140 000000
007142 015252
007144 015334

007146 104000
007150 014536
007152 104012
007154 000000
007156 015242
007160 015266

```
*****  
:GAIN ACCURACY TEST  
*****  
  
:THE GAIN ACCURACY TEST REQUESTS FOR A DECIMAL CH. TO BE TYPED IN VIA  
:THE OPERATOR. IT THEN REQUESTS '16' SPECIFIED VOLTAGES TO BE APPLIED TO  
:THAT CHANNEL. AFTER SUPPLY THE REQUESTED VOLTAGE, THE OPERATOR TYPES A  
:SPACE ON THE TELETYPE AND A SERIES OF '512' CONVERSIONS ARE THEN TAKEN  
:AT SPECIFIED GAIN SETTINGS AND AVERAGED OUT. IF THE AVERAGED VALUE  
:IS FOUND TO BE MORE THAN + OR -1 COUNT FROM THE INPUT VOLTAGES TRUE  
:VALUE, IT IS CONSIDERED IN ERROR AND THE ERROR VOLTAGE, THE EXPECTED VALUE  
:AND THE GAIN ARE TYPED OUT. AFTER COMPUTING INPUT VOLTAGE AVERAGES, A  
:TABLE OF ALL CONVERSION RESULTS ARE TYPED OUT.  
  
GAIN:  MOV      #GTO,AVECTR      ;SET UP '1A' RETURN ADDRESS  
        PRINT  
        MES3      ;TEXT GAIN ACCURACY TEST  
GTO:    PRINT  
        MES22     ;TEXT 'CH.?'  
        TTYIN     ;WAIT FOR REPLY  
        DECOCT    ;CONVERT TO OCTAL  
        MOVB      BCDTAB,ADWRD2+1 ;SET UP SELECTED CH.  
        CLR       MESPRT         ;CLR PRINT INHIBIT SWITCH  
  
:TEST +5.0V X G1  
GT1:    PRINT  
        MES23     ;TEXT '+5.00V'  
        WAITGN    ;CALL THE GAIN 'WAIT' HANDLER  
        G1        ;GAIN X1  
        POSS00  
        GP50X1    ;SAVE VALUE  
        TST       ADSIGN  
        BEQ       GT2           ;BRANCH TO NEXT TEST IF UNIPOLAR.  
  
:TEST -5.0V X G1  
        PRINT  
        MES24     ;TEXT 'SWITCH VOLTAGE NEG.'  
        WAITGN  
        G1        ;GAIN X1  
        NEG500    ;SHOULD=-5.0V  
        GM50X1    ;SAVE VALUE  
  
:TEST +2.5V X G1  
GT2:    PRINT  
        MES25     ;TEXT '+2.5V'  
        WAITGN  
        G1        ;GAIN X1  
        POS250    ;SAVE VALUE  
        GP25X1
```


1447				;TEXT +2.5V X G2 (5.0V)	
1448	007162	104013		TAKEGN	
1449	007164	000010		G2	;GAIN X2
1450	007166	015240		POS500	;SHOULD=+5.0V
1451	007170	015276		GP25X2	;SAVE VALUE
1452	007172	005767	005700	TST ADSIGN	
1453	007176	001414		BEQ GT3	;BRANCH IF UNIPOLAR
1454				;TEST -2.5V X G1	
1455	007200	104000		PRINT	
1456	007202	014521		MES24	;TEXT SWITCH VOLTAGE NEG.
1457	007204	104012		WAITGN	
1458	007206	000000		G1	;GAIN X1
1459	007210	015254		NEG250	
1460	007212	015336		GM25X1	;SAVE VALUE
1461					
1462				;TEST -2.5V X G2	
1463	007214	006267	005702	ASR KSTOR1	;=TO -5.0V
1464	007220	104013		TAKEGN	
1465	007222	000010		G2	;GAIN X2
1466	007224	015252		NEG500	
1467	007226	015346		GM25X2	;SAVE VALUE
1468					
1469				;TEST +1.25V X G1	
1470	007230	104000		GT3: PRINT	
1471	007232	014545		MES26	;TEXT '+1.25V'
1472	007234	104012		WAITGN	
1473	007236	000000		G1	;GAIN X1
1474	007240	015244		POS125	
1475	007242	015270		GP12X1	;SAVE VALUE
1476					
1477				;TEST '+1.25V X G2	
1478	007244	104013		TAKEGN	
1479	007246	000010		G2	;GAIN X2
1480	007250	015242		POS250	;=TO +2.5V
1481	007252	015300		GP12X2	;SAVE VALUE
1482				;TEST '+1.25 X G4	
1483	007254	104013		TAKEGN	
1484	007256	000020		G4	;GAIN X4
1485	007260	015240		POS500	
1486	007262	015310		GP12X4	;SAVE VALUE
1487	007264	005767	005606	TST ADSIGN	
1488	007270	001416		BEQ GT4	;BRANCH IF UNIPOLAR
1489					
1490				;TEST '-1.25V X G1	
1491	007272	104000		PRINT	
1492	007274	014521		MES24	;TEXT 'SWITCH VOLTAGE NEG.'
1493	007276	104012		WAITGN	
1494	007300	000000		G1	;GAIN X1
1495	007302	015256		NEG125	
1496	007304	015340		GM12X1	;SAVE VALUE
1497					

1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550

005500

```

:TEST -1.25V X G2
TAKEGN
G2
NEG250
GM12X2
:GAIN X2
:SAVE VALUE

:TEST -1.25V X G4
TAKEGN
G4
NEG500
GM12X4
:GAIN X4
:SHOULD = 5.0V
:SAVE VALUE

:TEST +0.625V X G1
GT4: PRINT
MES27
WAITGN
G1
POS625
GP62X1
:TEST +0.625V X G2
TAKEGN
G2
POS125
GP62X2
:GAIN X2
:SHOULD = +1.25V
:SAVE IT

:TEST +0.625V X G4
TAKEGN
G4
POS250
GP62X4
:GAIN X4
:SHOULD = +2.5V
:SAVE IT

:TEST +0.625V X G8
TAKEGN
G8
POS500
GP62X8
:GAIN X8
:SHOULD = +5.00V
:SAVE IT

TST ADSIGN
BEG GTS
:BRANCH IF UNIPOLAR

:TEST -0.625V X G1
PRINT
MES24
WAITGN
G1
NEG625
GM62X1
:SWITCH VOLTAGE NEG.
:GAIN X1
:SHOULD = -0.625V

:TEST -0.625V X G2
TAKEGN
G2
NEG125
GM62X2
:GAIN X2
:SHOULD = -1.25V
:SAVE IT

```

```

1549
1550 007424 104013
1551 007426 000020
1552 007430 015254
1553 007432 015362
1554
1555
1556 007434 104013
1557 007436 000030
1558 007440 015252
1559 007442 015372
1560
1561
1562 007444 104000
1563 007446 014564
1564 007450 104012
1565 007452 000000
1566 007454 015250
1567 007456 015274
1568
1569
1570 007460 104013
1571 007462 000010
1572 007464 015246
1573 007466 015304
1574
1575
1576 007470 104013
1577 007472 000020
1578 007474 015244
1579 007476 015314
1580
1581
1582 007500 104013
1583 007502 000030
1584 007504 015242
1585 007506 015324
1586 007510 005767
1587 007514 001422
1588
1589
1590 007516 104000
1591 007520 014521
1592 007522 104012
1593 007524 000000
1594 007526 015262
1595 007530 015344
1596
1597
1598 007532 104013
1599 007534 000010
1600 007536 015260
1601 007540 015354

```

005362

```

:TEST -0.625V X G4
      TAKEGN
      G4
      NEG250
      GM62X4
      :GAIN X4
      :SHOULD = -2.5V

:TEST -0.625V X G8
      TAKEGN
      G8
      NEG500
      GM62X8
      :GAIN X8
      :SHOULD = -5.00V

:TEST +0.3125V X G1
GTS: PRINT
      MES28
      WAITGN
      G1
      POS312
      GP31X1
      :TEXT '+0.3125V'
      :GAIN X1
      :SHOULD = +0.3125V
      :SAVE IT

:TEST +0.3125V X G2
      TAKEGN
      G2
      POS625
      GP31X2
      :GAIN X2
      :SHOULD = +0.625V

:TEST +0.3125V X G4
      TAKEGN
      G4
      POS125
      GP31X4
      :GAIN X4
      :SHOULD = +1.25V

:TEST +0.3125V X G8
      TAKEGN
      G8
      POS250
      GP31X8
      :GAIN X8
      :SHOULD = +2.50V

      TST      ADSIGN
      BEQ      GT6
      :BRANCH IS UNIPOLAR

:TEST -0.3125V X G1
      PRINT
      MES24
      WAITGN
      G1
      NEG312
      GM31X1
      :TEXT 'SWITCH NEG.'
      :GAIN X1
      :SHOULD = -0.3125V

:TEST -0.3125V X G2
      TAKEGN
      G2
      NEG625
      GM31X2
      :GAIN X2
      :SHOULD = -0.625V

```

1603
 1604 007542 104013
 1605 007544 000020
 1606 007546 015256
 1607 007550 015364
 1608
 1609
 1610 007552 104013
 1611 007554 000030
 1612 007556 015254
 1613 007560 015374
 1614
 1615
 1616 007562 104000
 1617 007564 014575
 1618 007566 104012
 1619 007570 000010
 1620 007572 015250
 1621 007574 015306
 1622
 1623
 1624 007576 104013
 1625 007600 000020
 1626 007602 015246
 1627 007604 015316
 1628
 1629
 1630 007606 104013
 1631 007610 000030
 1632 007612 015244
 1633 007614 015326
 1634 007616 005767
 1635 007622 001416
 1636
 1637
 1638 007624 104000
 1639 007626 014521
 1640 007630 104012
 1641 007632 000010
 1642 007634 015262
 1643 007636 015356
 1644
 1645
 1646 007640 104013
 1647 007642 000020
 1648 007644 015260
 1649 007646 015366
 1650
 1651
 1652 007650 104013
 1653 007652 000030
 1654 007654 015256
 1655 007656 015376
 1656

005254

```

:TEST -0.3125V X G4
      TAKEGN
      G4
      NEG125
      GM31X4
:GAIN X4
:SHOULD = -1.25V

:TEST -0.3125V X G8
      TAKEGN
      G8
      NEG250
      GM31X8
:GAIN X8
:SHOULD = -2.50V

:TEST +0.1563V X G2
GT6: PRINT
      MES29
      WAITGN
      G2
      POS312
      GP15X2
:TEXT '+0.1563V'
:GAIN X2
:SHOULD = +0.3125V

:TEST +0.1563V X G4
      TAKEGN
      G4
      POS625
      GP15X4
:GAIN X4
:SHOULD = +0.625V

:TEST 0.1563V X G8
      TAKEGN
      G8
      POS125
      GP15X8
:GAIN X8
:SHOULD = +1.25V

TST ADSIGN
BEQ GT7
:BRANCH IF UNIPOLAR

:TEST -0.1563V X G2
      PRINT
      MES24
      WAITGN
      G2
      NEG312
      GM15X2
:TEXT 'SWITCH NEG.'
:GAIN 'X2'
:SHOULD = -0.3125V

:TEST -0.1563V X G4
      TAKEGN
      G4
      NEG625
      GM15X4
:GAIN X4
:SHOULD = -0.625V

:TEST -0.1563V X G8
      TAKEGN
      G8
      NEG125
      GM15X8
:GAIN X8
:SHOULD = -1.25V

```

1657
 1658 007660 104000
 1659 007662 014606
 1660 007664 104012
 1661 007666 000020
 1662 007670 015250
 1663 007672 015320
 1664
 1665
 1666 007674 104013
 1667 007676 000030
 1668 007700 015246
 1669 007702 015330
 1670 007704 005767
 1671 007710 001412
 1672
 1673
 1674 007712 104000
 1675 007714 014521
 1676 007716 104012
 1677 007720 000020
 1678 007722 015262
 1679 007724 015370
 1680
 1681
 1682 007726 104013
 1683 007730 000030
 1684 007732 015260
 1685 007734 015400
 1686
 1687
 1688 007736 104000
 1689 007740 014617
 1690 007742 104012
 1691 007744 000030
 1692 007746 015250
 1693 007750 015332
 1694 007752 005767
 1695 007756 001406
 1696
 1697
 1698 007760 104000
 1699 007762 014521
 1700 007764 104012
 1701 007766 000030
 1702 007770 015262
 1703 007772 015402

005166

005120

:TEST +0.0781V X G4
 GT7: PRINT
 " 30
 TGN
 G4
 POS312
 GPO7X4

:TEXT '+0.0781V'
 :GAIN X4
 :SHOULD = +0.3125V

:TEST +0.0781V X B
 TAKEGN
 GB
 POS625
 GPO7X8
 TST ADSIGN
 BEQ GT8

:GAIN X8
 :SHOULD = +0.625V

:TEST -0.0781V X G4
 PRINT
 MES24
 WAITGN
 G4
 NEG312
 GPO7X4

:TEXT 'SWITCH NEG.'
 :GAIN X4
 :SHOULD = -0.3125V

:TEST -0.0781V X G8
 TAKEGN
 GB
 NEG625
 GPO7X8

:GAIN X8
 :SHOULD = -0.625V

:TEST +0.0390V X G8
 GT8: PRINT
 MES31
 WAITGN
 GB
 POS312
 GPO3X8
 TST ADSIGN
 BEQ GT9

:TEXT '+0.0390V'
 :GAIN X8
 :SHOULD = +0.3125V

:TEST -0.0390V X G8
 PRINT
 MES24
 WAITGN
 GB
 NEG312
 GPO3X8

:TEXT 'SWITCH NEG.'
 :SHOULD = -0.3125V

```

1704
1705 ;TYPE OUT A HISTOGRAM OF ALL THE GAIN AVERAGES.
1706
1707 007774 012767 015264 000142 GT9: MOV #GP5DX1,AVGTAB ;SET UP GAIN TABLE
1708 010002 012767 000002 005112 MOV #2,KSTOR1
1709 010010 104000 PRINT
1710 010012 014630 MES32 ;TYPE TABLE 'HEADER'.
1711 010014 012767 000005 003372 GT10: MOV #5,ICOUNT ;SET UP PRINT ROUTINE
1712 010022 104000 PRINT
1713 010024 014765 MES34 ;TYPE GAIN X1 VALUES
1714 010026 104014 PRTAVG ;TYPE OUT AVERAGES X1
1715 010030 104000 PRINT
1716 010032 014775 MES35 ;TYPE GAIN X2
1717 010034 012767 000001 005056 MOV #1,COUNT
1718 010042 104015 SIXDSH ;TYPE DASHES
1719 010044 104014 PRTAVG ;TYPE OUT AVERAGES X2
1720 010046 104000 PRINT
1721 010050 015005 MES36
1722 010052 012767 000002 005040 MOV #2,COUNT
1723 010060 104015 SIXDSH ;TYPE DASHES
1724 010062 104014 PRTAVG ;TYPE OUT AVERAGES X4
1725 010064 104000 PRINT
1726 010066 015015 MES37
1727 010070 012767 000003 005022 MOV #3,COUNT
1728 010076 104015 SIXDSH
1729 010100 104014 PRTAVG ;TYPE OUT AVERAGES X8
1730 010102 005767 004770 TST ADSIGN
1731 010106 001002 BNE GT11 ;IF UNIPOLAR, TYPE OUT NEG. COUNTS
1732 010110 000167 176752 JMP GT0 ;OTHERWISE RESTART GAIN TEST
1733 010114 005367 005002 GT11: DEC KSTOR1
1734 010120 001002 BNE +6
1735 010122 000167 176740 JMP GT0
1736 010126 104000 PRINT
1737 010130 014045 CRLF
1738 010132 000733 BR GT10
1739 ;SUBROUTINE TO TYPE OUT '5' AVERAGES FOR THE GAIN TEST HISTOGRAM.
1740
1741 010134 016767 003254 004774 XPRTAV: MOV ICOUNT,TEMP3
1742 010142 104010 XPTA1: PRTACT ;PRINT OCTAL VALUE OF GAIN AVERAGE
1743 010144 015264 AVGTAB: GP5DX1
1744 010146 062767 000002 177770 ADD #2,AVGTAB ;UPDATE GAIN TABLE
1745 010154 012767 000002 000644 MOV #2,SPACEX
1746 010162 104007 SPACE ;TYPE '2' SPACES
1747 010164 005367 004746 DEC TEMP3
1748 010170 001364 BNE XPTA1 ;IF NOT DONE, PRINT NEXT AVG.
1749 010172 000002 RTI
1750 ;SUBROUTINE TO TYPE OUT 'N' SETS OF SIX DASHES
1751
1752 010174 104000 DASH6: PRINT
1753 010176 015025 MES38
1754 010200 005367 004714 DEC COUNT
1755 010204 001373 BNE DASH6
1756 010206 000002 RTI

```

```

1757 010210 104011 XWATGN: TTYIN ;WAIT FOR 'CR' BEFORE CONTINUING
1758 010212 117667 MOV#B @ (SP),ADWRD2 ;GET SPECIFIED GAIN AND SET IT UP.
1759 010220 062716 ADD #2,(SP) ;SET UP THE ADDRESS OF TRUE VOLTAGE
1760 010224 017667 MOV @ (SP),PRTADR ;SAVE ADDRESS OF TRUE VOLTAGE
1761 010232 062716 ADD #2,(SP) ;SET UP STORAGE ADDRESS FOR VOLTAGE
1762 010236 017667 MOV @ (SP),KSTOR2 ;SAVE ADDRESS
1763 010244 062716 ADD #2,(SP) ;SET UP STACK TO EXIT
1764 010250 104016 GLOOP: TSTTKS ;TEST FOR KEYBOARD FLAG
1765 010252 012767 MOV #1000,COUNT ;SET UP TO TAKE '512' CONVERSIONS
1766 010260 004767 JSR PC,ADCNV ;TAKE THE CONVERSIONS
1767 010264 104004 COMPUTE ;COMPUTE THE AVERAGE
1768 010266 016777 MOV AVRAGE,@KSTOR2 ;SAVE THE AVERAGE VALUE
1769 010274 104005 CATORIZ ;CATAGORIZE THE COUNT SPREAD
1770 010276 026777 CMP AVRAGE,@PRTADR ;IS AVERAGE =TO KNOWN VALUE?
1771 010304 001414 BEQ GANEXT ;EXIT IF EQUAL
1772 010306 026777 CMP AVERP1,@PRTADR ;IS AVERAGE =TO KNOWN VALUE +1?
1773 010314 001410 BEQ GANEXT ;EXIT IF EQUAL
1774 010316 027767 CMP @PRTADR,AVERM1 ;IS AVERAGE =TO KNOWN VALUE -1?
1775 010324 001404 BEQ GANEXT ;EXIT IF EQUAL
1776 010326 032777 GAINER: BIT #SW14,@SWR ;TEST FOR INHIBIT SCOPE LOOPING
1777 010334 001401 BEQ GERR1 ;BRANCH IF NOT SET
1778 010336 000002 GANEXT: RTI
1779 010340 032777 GERR1: BIT #SW13,@SWR ;TEST FOR PRINT INHIBIT
1780 010346 001340 BNE GLOOP ;BRANCH IS SW SET
1781 010350 005767 TST MESPRT ;TEST IF TITLE HAS BEEN TYPED
1782 010354 001005 BNE GERR2 ;BRANCH IF YES
1783 010356 052767 BIS #1,MESPRT ;OTHERWISE TYPE ERROR HEADER
1784 010364 104000 PRINT
1785 010366 014735 MES33 ;TEXT 'GAIN'
1786 010370 105767 GERR2: TSTB ADWRD2 ;TEST FOR 'G1'
1787 010374 001003 BNE GERR3 ;BRANCH IF NOT
1788 010376 104000 PRINT
1789 010400 014765 MES34 ;TEXT '1'
1790 010402 000420 BR GERR6
1791 010404 122767 GERR3: CMPB #10,ADWRD2 ;TEST FOR GX2
1792 010412 001003 BNE GERR4 ;BRANCH IF NOT -
1793 010414 104000 PRINT
1794 010416 014775 MES35 ;TEXT '2'
1795 010420 000411 BR GERR6
1796 010422 122767 GERR4: CMPB #20,ADWRD2 ;TEST FOR GAIN OF '4'
1797 010430 001003 BNE GERR5 ;BRANCH AND PRINT GX8
1798 010432 104000 PRINT
1799 010434 015005 MES36 ;TEXT '4'
1800 010436 000402 BR GERR6
1801 010440 104000 PRINT
1802 010442 015015 MES37 ;TEXT '8'
1803 010444 1C-010 GERR6: PRTOCT
1804 010446 000000 PRTADR: 0 ;TYPE VOLTAGE VALUE
1805 010450 104007 SPACE ;TYPE SPACE
1806 010452 104007 SPACE
1807 010454 104010 PRTOCT
1808 010456 015162 AVRAGE ;TYPE AVERAGE
1809 010460 104000 PRINT
1810 010462 014045 CRLF
1811 010464 000671 BR GLOOP ;RETEST GAIN AVERAGE
1812

```

1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853

010466 012767 010506 004414
010474 104000
010476 014150
010500 012767 000010 002706
010506 104000
010510 014276
010512 104011
010514 104001
010516 016767 000714 004376
010524 016767 000710 004372
010532 104003
010534 116767 004334 004364
010542 016767 000672 000666
010550 104017
010552 116767 004316 004350
010560 104016
010562 012767 000010 004330
010570 116767 004326 004277
010576 116767 004324 004270
010604 004767 000636
010610 116767 004310 004257
010616 116767 004306 004250
010624 004767 000616
010630 032777 020000 004222
010636 001350
010640 104000
010642 014171
010644 016702 004254
010650 104006
010652 104007
010654 012767 015400 177262
010662 104014
010664 000735

```

:RECOVERY TEST
:*****
;THIS TEST IS DESIGNED TO TEST RECOVERY OF THE A/D CONVERTER VIA ACCEPT-
;ING TWO (2) CHANNEL AND GAIN INPUTS FROM THE TELETYPE AND THEN TAKE A
;SERIES OF EIGHT CONVERSIONS ON EACH CHANNEL AND TYPING OUT THE CON-
;VERSION VALUES OF THE 2ND CHANNEL IN THE ORDER THEY WERE TAKEN.

RECVRY: MOV      #RECVY1,AVECTR ;SET UP THE '1A' RETURN ADDRESS
        PRINT
        MES8          ;TEXT 'RECOVERY TEST'
RECVY1: MOV      #10,ICOUNT    ;SET UP TO PRINT '8' VALUES
        PRINT        ;REQUEST CHANNELS
        MES14
        TTYIN        ;WAIT FOR INPUT
        DECOCT       ;CONVERT TO OCTAL
        MOV      BCDTAB,KSTOR1 ;SAVE 1ST CH.
        MOV      BCDTAB+2,KSTOR2 ;SAVE 2ND CH.
        GAININ      ;GET THE 1ST GAIN SETTING
        MOV      ADWRD2,KSTOR3 ;SAVE IT
        MOV      BCDTAB+2,BCDTAB ;SET UP TO CONVERT 2ND GAIN
        GAINXN      ;DECODE 2ND GAIN
        MOV      ADWRD2,KSTOR4 ;SAVE IT
RECVY2: TSTTKS      ;CHECK FOR KEYBOARD FLAG
        MOV      #10,COUNT     ;SETUP TO TAKE '8' CONVERSIONS ON 1ST CH.
        MOV      KSTOR1,ADWRD2+1 ;LOAD 1ST CH.
        MOV      KSTOR3,ADWRD2   ;SET UP GAIN
        JSR      PC,ADCHVT      ;TAKE THE CONVERSIONS
        MOV      KSTOR2,ADWRD2+1 ;SET UP 2ND CH.
        MOV      KSTOR4,ADWRD2   ;LOAD 2ND GAIN
        JSR      PC,ADCHVT      ;TAKE 2ND SERIES OF CONVERSIONS
        BIT      #SW13,JSWR      ;TEST THE PRINT INHIBIT SW
        BNE      RECVY2        ;BRANCH IF SET
        PRINT
        MES9          ;TEXT 'CH.'
        MOV      KSTOR2,R2
        BINDEC
        SPACE
        MOV      #ADBUFF,AVGTAB
        PRTAVG      ;PRINT VALUES OF 2ND CH.
        BR RECVY2    ;DO IT AGAIN
    
```



```

1854 ;ROUTINE TO LOOP THUR A SINGLE LOGIC SUBTEST. ENTERED FROM THE 'MONITOR'
1855 ;VIA TYPING 'TNN' WHERE 'NN' IS EQUATED TO THE 'PC' OF A SUBTEST.
1856 ;NOTE THAT 'SW11' MUST BE '0' (DOWN) TO RUN THIS TEST.
1857
1858 010666 004767 170600 TESTX: JSR PC, SETUP ;MAIN INITIALIZATION FOR LOGIC TEST
1859 010672 012701 011272 MOV #INBUF, R1
1860 010676 005067 004220 CLR KSTOR1
1861 010702 042711 177770 TSTA: BIC #177770, (R1) ;STRIPE NO. TO OCTAL
1862 010706 062167 004210 ADD (R1)+, KSTOR1 ;ADD TO LAST RESULT
1863 010712 005367 004200 DEC CHRCNT
1864 010716 001407 BEQ TSTXB
1865 010720 006367 004176 ASL KSTOR1
1866 010724 006367 004172 ASL KSTOR1
1867 010730 006367 004166 ASL KSTOR1
1868 010734 000762 BR TSTA
1869
1870 010736 022767 005754 004156 TSTXB: CMP #TSTDN, KSTOR1 ;IS NO. WITHIN LIMITS OF THE LOGIC TEST
1871 010744 100002 BPL +6 ;CONTINUE IF YES
1872 010746 000167 170512 JMP INIT3 ;OTHERWISE RETURN TO MONITOR
1873 010752 062767 000002 004142 ADD #2, KSTOR1 ;ADD '2' TO PCINT TO INSTRUCTION AFTER SCOPE
1874 010760 005067 002432 XLOOP: CLR SCOPEF ;KEEP COUNT AT ZERO
1875 010764 012767 010760 002426 MOV #XLOOP, RETURN ;LOAD SCOPE LOOP RETURN POINTER
1876 010772 000177 004124 JMP @KSTOR1 ;JUMP TO TEST
1877
1878 ;SUBROUTINE TO ISSUE N SPACES
1879 ;N IS ONE PLUS VALUE CONTAINED IN SPACEX
1880 ;SPACEX IS CLEARED WITHIN THE SUBROUTINE, SO THAT A CALL ON
1881 ;SPACE WITHOUT LOADING SPACEX ISSUES ONLY ONE SPACE
1882 010776 105777 004052 XSPACE: TSTB @TPS ;WAIT FOR TTY READY
1883 011002 100375 BPL -4
1884 011004 012777 000240 004044 MOV #240, @TPB ;OUTPUT A SPACE
1885 011012 005367 000010 DEC SPACEX ;DECREMENT COUNT
1886 011016 003367 BGT XSPACE ;LOOP IF NOT DONE
1887 011020 005067 000002 CLR SPACEX ;RESET COUNT TO ZERO
1888 011024 000002 RTI ;RETURN
1889 011026 000000 SPACEX: 0

```

```

1890                                     ;KEYBOARD SERVICE ROUTINE
1891
1892 011030 012704 011272 XTTYIN: MOV      #INBUF,R4      ;SETUP CHARACTER BUFFER
1893 011034 005067 004056 NEWIN: CLR      CHRCNT      ;CLEAR CHARACTER COUNTER
1894 011040 105777 004004 INPUTA: TSTB    #TKS        ;CHARACTER READY?
1895 011044 100375          BPL      INPUTA      ;NO, WAIT IT OUT
1896 011046 017701 004000          MOV      #TKB,R1      ;SAVE CHARACTER
1897 011052 042701 000200          BIC      #200,R1     ;STRIPE PARITY BIT
1898 011056 120127 000060          CMPB    R1,#60      ;IS IT A SPECIAL CHARACTER
1899 011062 100427          BMI      SPCHR      ;YES, TEST IT
1900 011064 122701 000137          CMPB    #137,R1
1901 011070 100424          BMI      SPCHR
1902 011072 122701 000124          CMPB    #124,R1      ;IS CHAR 'T'
1903 011076 001004          BNE      INPUTB      ;NO
1904 011100 012767 000372 003772          MOV      #372,SUBX    ;YES, SET THE SUBTEST CALL SW.
1905 011106 000407          BR      OUTPTA      ;IGNORE 'T' AND SAVE ADDRESS
1906 011110 010124          INPUTB: MOV     R1,(R4)+ ;SAVE CHARACTER
1907 011112 005267 004000          INC      CHRCNT      ;INCREMENT THE CHARACTER COUNT.
1908 011116 022767 000011 003772          CMP      #11,CHRCNT
1909 011124 100457          BMI      SPCHRS      ;TYPE '?' IF TOO MANY CHAR.
1910 011126 105777 003722          OUTPTA: TSTB    #TPS      ;ECHO CHARACTER
1911 011132 100375          BPL      OUTPTA
1912 011134 003177 003716          MOVB    R1,#TPB
1913 011140 000737          BR      INPUTA      ;WAIT FOR NEXT CHARACTER
1914                                     ;SUBROUTINE TO TEST FOR SPECIAL CHARACTERS : 'A', 'C', '+', 'CR', '.', ' OR 'RUBOUT'
1915
1916 011142 122701 000001          SPCHR:  CMPB    #1,R1      ;CHAR. = 'A'
1917 011146 001005          BNE      SPCHR1      ;NO, NOT 'A'
1918 011150 104000          PRINT   CNTRLA      ;ECHO 'A'
1919 011152 014042          CNTRLA POP2SP      ;RESTORE 'SP'
1920 011154 022626          POP2SP
1921 011156 000177 003726          JMP      @AVECTR      ;YES, EXIT VIA 'A' VECTOR ADDRESS.
1922 011162 122701 000003          SPCHR1: CMPB    #3,R1      ;CHAR. = 'C'
1923 011166 001002          BNE      +6          ;NO, NOT 'C'
1924 011170 000167 170074          JMP      MONITR      ;YES, EXIT TO MONITOR
1925 011174 122701 000177          CMPB    #177,R1     ;CHAR. = 'RUBOUT'
1926 011200 001011          BNE      SPCHR2      ;IGNORE CHAR. & EXIT
1927 011202 005767 003710          TST     CHRCNT      ;IS RUBOUT LEGAL?
1928 011206 001714          BEQ     INPUTA      ;NO, IGNORE IT
1929 011210 005367 003702          JEC     CHRCNT
1930 011214 012701 000134          MOV     #134,R1     ;TYPE '\ ' TO INDICATE RUBOUT
1931 011220 005744          TST     -(R4)       ;POP OFF LAST CHARACTER
1932 011222 000741          BR      OUTPTA      ;WAIT FOR NEXT CHARACTER
1933 011224 122701 000053          SPCHR2: CMPB    #53,R1    ;TEST FOR '+'
1934 011230 001004          BNE      SPCHR3      ;BRANCH IF NO
1935 011232 012767 000177 003636          MOV     #177,ADSIGN  ;YES, INDICATES UNIPOLAR
1936 011240 000732          BR      OUTPTA      ;WAIT NEXT CHAR.
1937 011242 122701 000054          SPCHR3: CMPB    #54,R1    ;TEST FOR '.'
1938 011246 001720          BEQ     INPUTB      ;LEGAL CHAR., SAVE IT
1939 011250 122701 000015          SPCHR4: CMPB    #15,R1   ;=TO 'CARRIAGE RETURN' TO TERMINATE'
1940 011254 001003          BNE      SPCHRS      ;NO, CONTINUE
1941 011256 104000          PRINT   CR LF      ;YES, TYPE 'CR-LF'
1942 011260 014045          CR LF
1943 011262 000002          EXTTY:  RTI          ;EXIT

```

```

:944 011264 104000
:945 011266 014052
:946 011270 000661
:947 011272 000000
:948 011306 011306
:949
:950
:951
:952 011306 012704 011272
:953 011312 012703 011436
:954 011316 005067 000116
:955 011322 005001
:956 011324 005002
:957 011326 005767 003564
:958 011332 003426
:959 011334 005367 003556
:960 011340 122714 000054
:961 011344 001421
:962 011346 121427 000060
:963 011352 002424
:964 011354 021427 000071
:965 011360 003021
:966 011362 042714 177760
:967 011366 012400
:968 011370 010102
:969 011372 006301
:970 011374 006301
:971 011376 006301
:972 011400 060201
:973 011402 060201
:974 011404 060001
:975 011406 000747
:976 011410 005724
:977 011412 010123
:978 011414 005767 003476
:979 011420 001340
:980 011422 000002
:981 011424 104000
:982 011426 014052
:983 011430 104011
:984 011432 000167 177650
:985 011436 000000
:986 011440 000000
:987 011442 000000
:988 011444 000000
    
```

```

SPCHRS: PRINT ; OTHERWISE TYPE '?'
          QMARK
          BR NEWIN ; WAIT FOR NEW ENTRY
INBUF: 0 ; CHARACTER STORAGE BUFFER
          .=.+12
; SUBROUTINE WILL CONVERT 'N' BCD WORDS (SEPARATED VIA COMMA'S)
; WHICH WERE STORED IN A TABLE VIA 'TTYIN' TO OCTAL AND STORE THEM.

BCDBIN: MOV #INBUF,R4 ; SETUP ASCII STORAGE TABLE
          MOV #BCDTAB,R3 ; TABLE FOR STORAGE OF CONVERTED WORDS
          CLR BCDTAB+2
BCDBN1: CLR R1 ; REG. TO STORE RUNNING TOTAL
          CLR R2 ; TEMP. STORAGE FOR 'R1'
BCDBN2: TST CHRCNT ; END OF DATA?
          BLE BCDEND ; YES, EXIT
          DEC CHRCNT ; DECREMENT CHARACTER COUNTER
          CMPB #54,(R4) ; IS CHARACTER = TO '?'
          BEQ BCDEND ; YES, DECODE NEW WORD
          CMPB (R4),#60 ; TEST FOR LEGAL NO.
          BLT BCDERR
          CMP (R4),#71
          BGT BCDERR
          BIC #177760,(R4) ; STRIPE NO. TO BCD
          MOV (R4)+,R0 ; SAVE NO. IN R0
          MOV R1,R2 ; SAVE CURRENT TOTAL
          ASL R1 ; NX2
          ASL R1 ; NX4
          ASL R1 ; NX8
          ADD R2,R1 ; NX9
          ADD R0,R1 ; NX10
          BR BCDNB2 ; N+NEW NO.

BCDEND: TST (R4)+ ; UPDATE BUFFER
          MOV R1,(R3)+ ; SAVE CONVERTED VALUE & SETUP TO SAVE NEXT
          TST CHRCNT ; FINISHED?
          BNE BCDNB1 ; NO, CONVERT NEXT WORD
          RTI ; YES, EXIT

BCDERR: PRINT ; TYPE '?'
          QMARK ; TO BE TYPED ON QUESTIONABLE ENTRIES.
          TTYIN
          JMP BCDBIN

BCDTAB: 0 ; OCTAL STORAGE TABLE
          0
          0
          0
    
```

```

1989 ;SUBROUTINE TO TAKE 'N' CONVERSIONS AND STORE THEM IN AN A/D BUFFER. ROUTINE
1990 ;IS ENTERED WITH 'N' IN COUNT AND THE CH & GAIN TO BE CONVERTED IN 'ADWORD'.
1991
1992 011446 016767 003446 003456 ADCNVT: MOV COUNT,TEMP1 ;SET UP NO. OF CONVERSIONS TO BE TAKEN
1993 011454 012704 015400 ;MOV #ADBUFF,R4 ;SET UP BUFFER ADDRESS.
1994 011460 005777 003404 ;TST @ADDBR ;CLR A/D DONE
1995 011464 116777 003404 003372 ;MOV# ADWRD2,@ADCS ;SET UP GAIN AND EXT CONTROL IF SET
1996 011472 116777 003377 003366 CNVT1: MOV# ADWRD2+1,@ADCS0 ;LOAD CH. & START CONVERT IF NOT 'EXT'
1997 011500 105777 003360 ;TSTB @ADCS ;TEST A/D DONE
1998 011504 100375 ;BPL -4 ;WAIT FOR DONE
1999 011506 005777 003352 ;TST @ADCS ;TEST ERROR BIT
2000 011512 100002 ;BPL CNVT2 ;BRANCH IF NOT SET
2001 011514 104000 ;PRINT ;OTHERWISE TYPE ERROR
2002 011516 014230 ;MES12 ;TEXT 'ERROR BIT SET'.
2003 011520 017724 003344 CNVT2: MOV @ADDBR,(R4)+ ;SAVE DATA
2004 011524 005367 003402 ;DEC *TEMP1 ;DECREMENT COUNTER
2005 011530 003360 ;BGT CNVT1 ;IF NOT '0' TAKE NEXT CO' ERSION
2006 011532 000207 ;RTS PC
2007
2008 ;POWER FAIL HANDLER
2009 011534 010046 PWRFAL: MOV R0,-(SP)
2010 011536 010146 ;MOV R1,-(SP)
2011 011540 010246 ;MOV R2,-(SP)
2012 011542 010346 ;MOV R3,-(SP)
2013 011544 010446 ;MOV R4,-(SP)
2014 011546 010546 ;MOV R5,-(SP)
2015 011550 016746 166250 ;MOV R24,-(SP)
2016 011554 010667 003334 ;MOV SP,PROC
2017 011560 012767 011570 166236 ;MOV #PWRUP,24
2018 011566 000000 ;HALT
2019 ;POWER UP HANDLER
2020 011570 012777 000340 003250 PWRUP: MOV #340,@PSW
2021 011576 016706 003312 ;MOV PROC,SP
2022 011602 012667 166216 ;MOV (SP)+,24
2023 011606 012605 ;MOV (SP)+,R5
2024 011610 012604 ;MOV (SP)+,R4
2025 011612 012603 ;MOV (SP)+,R3
2026 011614 012602 ;MOV (SP)+,R2
2027 011616 012601 ;MOV (SP)+,R1
2028 011620 012600 ;MOV (SP)+,R0
2029 011622 000005 ;RESET
2030 011624 104000 ;PRINT
2031 011626 014462 ;MES21
2032 011630 000167 167434 ;JMP MONITR

```

```

2033
2034 ;SUBROUTINE TO INPUT A 'GAIN FROM THE TELETYPE AND STORE IT IN THE 'ADWORD'
2035
2036 011634 104000 XGAINA: PRINT
2037 011636 014330 MES18 ;TEXT 'GAIN?'
2038 011640 104011 TTYIN ;WAIT FOR INPUT
2039 011642 104001 DECOCT ;CONVERT TO OCTAL
2040 011644 022767 000001 177564 XGAINB: CMP #1,BCDTAB ;TEST FOR '1'
2041 011652 001003 BNE XGAIN2 ;IF NOT '1' TEST FOR '2'
2042 011654 105067 003214 CLRB ADWRD2 ;OTHERWISE SELECT GAIN X1
2043 011660 000002 RTI
2044 011662 022767 000002 177546 XGAIN2: CMP #2,BCDTAB
2045 011670 001004 BNE XGAIN4
2046 011672 112767 000010 003174 MOVB #10,ADWRD2
2047 011700 000002 RTI
2048 011702 022767 000004 177526 XGAIN4: CMP #4,BCDTAB
2049 011710 001004 BNE XGAIN8
2050 011712 112767 000020 003154 MOVB #20,ADWRD2
2051 011720 000002 RTI
2052 011722 022767 000010 177506 XGAIN8: CMP #10,BCDTAB
2053 011730 001341 BNE XGAINA ;ILLEGAL ENTRY, TRY AGAIN
2054 011732 112767 000030 003134 MOVB #30,ADWRD2
2055 011740 000002 RTI
2056

```

2057
2058
2059
2060
2061
2062
2063 011742 011646
2064 011744 162716 000002
2065 011750 017616 000000
2066 011754 005716
2067 011756 001001
2068 011760 000000
2069 011762 006316
2070 011764 042716 177001
2071 011770 062716 012002
2072 011774 017616 000000
2073 012000 000136
2074
2075
2076
2077 012002 012042
2078 012004 011306
2079 012006 013332
2080 012010 011634
2081 012012 012334
2082 012014 012526
2083 012016 012136
2084 012020 010776
2085 012022 013112
2086 012024 011030
2087 012026 010210
2088 012030 010212
2089 012032 010134
2090 012034 010174
2091 012036 013230
2092 012040 011644
2093

; EMT DISPATCH SERVICE ROUTINE
; ARGUMENT OF EMT IS EXTRACTED AND USED AS OFFSET TO OBTAIN POINTER
; TO THE SELECTED SUBROUTINE.

```
EMTSRV: MOV      (SP), -(SP)      ; GET PC FOR TO RETURN
          SUB      #2, (SP)      ; PC OF EMT
          MOV      @ (SP), (SP)  ; GET EMT
          TST      (SP)          ; IS EMT VALID?
          BNE      EMTOK
          HALT
EMTOK:   ASL      (SP)          ; INVALID EMT
          BIC      #177001, (SP) ; MULTIPLY EMT ARG BY '2'
          ADD      #EMTTAB, (SP) ; CLEAR UNWANTED BITS
          MOV      @ (SP), (SP)  ; POINTER TO SUBROUTINE ADDRESS
          JMP      @ (SP)+      ; SUBROUTINE ADDRESS
          ; GO TO SUBROUTINE
```

; EMT DISPATCH TABLE

```
EMTTAB: TYPMES      ; MESSAGE PRINT ROUTINE
          BCD BIN    ; DECIMAL TO BINARY CONVERSION ROUTINE
          SCOPEC     ; LOGIC TEST SCOPE ROUTINE
          XGAINA     ; REQUEST A 'GAIN' FROM THE TTY.
          CMPTE      ; SUBROUTINE TO COMPUTE THE AVG
          CATORZ     ; SUBROUTINE TO COMPUTE 'COUNT SPREAD'
          DECPRT     ; SUBROUTINE TO CONVERT OCT TO DEC + PRINT
          XSPACE     ; SUBROUTINE TO TYPE SPACES
          OCTPRT     ; OCTAL PRINT ROUTINE
          XTTYIN     ; TELEPRINTER SERVICE ROUTINE
          XWATGN     ; GAIN TEST CONVERSION ROUTINE
          XWATGN+2   ; GAIN TEST CONVERSION ROUTINE
          XPRTAV     ; SUBROUTINE TO PRINT OUT THE GAIN AVERAGES
          DASH6      ; SUBROUTINE TO TYPE OUT '6' DASHES
          TKSFLG     ; SUBROUTINE TO TEST FOR KEYBOARD FLAG
          XGAINB     ; SUBROUTINE TO DECODE A GAIN FOR TTY
```

: MESSAGE PRINT ROUTINE, ENTERED VIA EMT DISPATCH HANDLER.
: ROUTINE PICKS UP CONTENTS OF THE 'PC' AND USES THIS AS
: THE ADDRESS OF MESSAGE TO BE TYPED.

012042 005077 003000
012046 017605 000000
012050 062716 000002
012056 105777 002772
012062 100375
012064 122715 000100
012070 001001
012072 000002
012074 122715 000045
012100 001403
012102 112577 002750
012106 000763
012110 012777 000015 002740
012116 105777 002732
012122 100375
012124 012777 000012 002724
012132 105725
012134 00075C

012136 005077 002704
012142 012767 177774 000146
012150 012767 012324 000144
012156 012767 000240 000134
012164 012767 :77777 000122
012172 005267 000116
012176 167702 000120
012202 100373
012204 067702 000112
012210 004767 000020
012214 005267 000076
012220 001001
012222 000002
012224 062767 000002 000070
012232 000754
012234 005767 000054
012240 00:010
012242 022757 177777 000046
012250 001404
012252 016767 000042 000034
012260 000406
012262 012767 000260 000030
012270 052767 000260 000016
012276 105777 002552
012302 100375
012304 016777 000004 002544
012312 000207

TYPMES: CLR @PSW ; ENABLE KEYBOARD INT.
MOV @ (SP), R5 ; GET THE MESSAGE ADDRESS FROM START
AND @2 (SP) ; SET UP STACK TO EXIT
TYPERA: TSTB @TPS
BPL TYPERA ; WAIT FOR TTY DONE
CMPB #100 (R5) ; TEST FOR '0'
BNE TYPERA1 ; BRANCH IF NO EQUAL
RTI ; OTHERWISE EXIT
TYPERA1: CMPB #45 (R5) ; TEST FOR '%'
BEQ TYPECL ; IF = TYPE 'CR-LF'
TYPERA2: MOVB (R5)+, @TPB ; OUTPUT CHAR
BR TYPERA
TYPECL: MOV #15, @TPB ; TYPE 'CR'
TSTB @TPS
BPL -4
MOV #12, @TPB ; INCREMENT BUFFER
TSTB (R5)+
BR TYPERA

: PRINT DECIMAL VALUE IN R2

DECPRT: CLR @PSW ; ENABLE TTY INTERRUPTS
MOV #-4, DIGCNT
MOV #DECPNT+2, DECPNT
MOV #240, ZERO
TYPT1: MOV #-1, DIGIT
TYPT2: INC DIGIT
SJB @DECPNT,
BPL TYPT2
ADD @DECPNT, %2
JSR PC, DECOU
INC DIGCNT
BNE TYPT3
RTI
TYPT3: ADD #2, DECPNT
BR TYPT1
DECOU: TST DIGIT
BNE DECI
CMP #-1, DIGCNT
BEQ DECI
MOV ZERO, DIGIT
RR DECI
DEC1: MOV #260, ZERO
BIS #260, DIGIT
DECI: TSTB @TPS
BPL -4
MOV DIGIT, @TPB
RYS 7

```

2147 012314 000000
2148 012316 000000
2149 012320 000240
2150 012322 012324
2151 012324 001750
2152 012326 000144
2153 012330 000012
2154 012332 000001
2155
2156
2157
2158 012334 012767 000777 002570
2159 012342 005067 002574
2160 012346 012704 015400
2161 012352 012467 002604
2162 012356 016767 002600 002560
2163 012364 016767 002572 002554
2164 012372 066767 002514 002562
2165 012400 012467 002530
2166 012404 026767 002524 002532
2167 012412 003403
2168 012414 116767 002514 002522
2169 012422 126767 002506 002516
2170 012430 003003
2171 012432 016767 002476 002506
2172 012440 066767 002446 002466
2173 012446 066767 002462 002506
2174 012454 005367 002462
2175 012460 005367 002446
2176 012464 001345
2177 012466 012767 000011 002436
2178 012474 006267 002442
2179 012500 006067 002456
2180 012504 005367 002422
2181 012510 001371
2182 012512 005567 002444
2183 012516 166767 002370 002436
2184 012524 000002
  
```

```

DIGIT: 0
DIGCNT: 0
ZERO: 240
DECPNT: .+2
          1000.
          100.
          10.
          1.
  
```

; COMPUTE THE RESULTS OF 512 CONVERSIONS AS HIGH, LOW AND AVERAGE

```

CMPTE: MOV #777,TEMP1 ;SET UP TO COMPARE '511' NUMBERS
        CLR HIORDV ;CLR HI ORDER DIVIDEND
        MOV #A0BUFF,R4 ;SET UP DATA BUFFER ADDRESS
        MOV (R4)+,AVRAGE ;STORE 1ST VALUE AS AVERAGE
        MOV AVRAGE,HIGH ;HIGH
        MOV AVRAGE,LOW ;& LOW
        ADD ADSIZE,AVRAGE ;ADD OFFSET TO AVERAGE
GETDAT: MOV (R4)+,TEMP2
        CMP TEMP2,HIGH ;IS NEW NO. GREATER THAN OLD NO.
        BLE TSLO ;BRANCH IF NOT GREATER
        MOV TEMP2,HIGH ;OTHERWISE SAVE AS NEW HIGH
TSLC: CMP TEMP2,LOW
        BGT TAGA ;OTHERWISE SAVE AS NEW LOW
TAGA: MOV TEMP2,LOW ;ADD OFFSET TO MAKE ALL NO. POS.
        ADD ADSIZE,TEMP2 ;ADD LOW ORDER
        ADD TEMP2,AVRAGE ;ADD CARRY TO HI ORDER
        ADC HIORDV
        DEC TEMP1
        BNE GETDAT ;512 ADDITIONS?
AVGDAT: MOV #11,TEMP1 ;YES, DIVIDE/512
        ASR HIORDV ;SHIFT CARRY BIT INTO LO ORDER
        ROR AVRAGE
        DEC TEMP1
        BNE AVGDAT ;DONE?
        ACC AVRAGE ;YES, ADD REMAINDER TO LO ORDER
        SUB ADSIZE,AVRAGE ;SUBTRACT OFFSET *0 OBTAIN REAL AVERAGE
  
```



```

2185                                     :SUBROUTINE TO CALCULATE THE PLUS & MINUS 5 COUNT LIMITS FROM AN AVERAGE
2186
2187 012526 012767 000005 002376 CATRZ: MOV      #5,TEMP1
2188 012534 016767 002422 002372      MOV      AVERAGE,TEMP2      :MOV AVER. TO WORK AREA
2189 012542 066767 002344 002364      ADD      ADSIZE,TEMP2      :MAKE AVG. POS.
2190 012550 012703 015164      MOV      #AVERP1,R3      :SETUP DISTRIBUTION TABLE (POS.)
2191 012554 005267 002354      FILE1: INC      TEMP2      :A=A+1
2192 012560 016713 002350      MOV      TEMP2,(R3)      :SAVE A+1
2193 012564 166723 002322      SUB      ADSIZE,(R3)+    :RESTORE ORIGINAL VALUE
2194 012570 005367 002336      DEC      TEMP1          :SAVED '5' COUNTS?
2195 012574 001367      BNE      FILE1          :BRANCH IF NO
2196
2197                                     :SET UP TABLE OF AVG. -1 TO -5
2198 012576 012767 000005 002326      MOV      #5,TEMP1
2199 012604 016767 002352 002322      MOV      AVERAGE,TEMP2    :MOV AVG. TO WORK AREA.
2200 012612 066767 002274 002314      ADD      ADSIZE,TEMP2
2201 012620 012703 015162      MOV      #AVERAGE,R3      :SET UP DISTRIBUTION TABLE NEG.
2202 012624 005367 002304      FILE2: DEC      TEMP2      :A=-1
2203 012630 016743 002300      MOV      TEMP2,-(R3)      :SAVE 'A-1'
2204 012634 166713 002252      SUB      ADSIZE,(R3)      :RESTORE ORIGINAL NO. -1
2205 012640 005367 002266      DEC      TEMP1          :SAVED '5' COUNTS?
2206 012644 001367      BNE      FILE2          :BRANCH IF NO
2207
2208                                     :CATEGORIZE THE COUNT SPREAD AS '+6 & -6' COUNTS FROM THE AVERAGE
2209 012646 012703 015176      CATR1: MOV      #ORLOW,R3      :CLEAR COUNTS
2210 012652 J05023      CLR      (R3)+
2211 012654 022703 015230      CMP      #ORHIGH+2,R3     :FINISHED?
2212 012660 001374      BNE      CATR1          :NO, CLEAR NEXT COUNTER
2213 012662 012767 001001 002242      MOV      #1001,TEMP1      :COMPARE '512' COUNTS
2214 012670 012700 015400      MOV      #ADBUFF,R0      :SET JP A/D BUFFER
2215 012674 005367 002232      CATR2: DEC      TEMP1
2216 012700 001437      BEQ      CATR5          :EXIT IF '0'
2217 012702 012067 002226      MOV      (R0)+,TEMP2
2218 012706 026767 002262 002220      CMP      AVERP5,TEMP2
2219 012714 100423      BMI     OVRHI
2220 012716 026767 002212 002224      CMP      TEMP2,AVERM5
2221 012724 100422      BMI     OVRLO
2222 012726 005001      CLR      R1
2223 012730 012702 015150      MOV      #AVERM5,R2
2224 012734 022267 002174      CATR3: CMP      (R2)+,TEMP2
2225 012740 001405      BEQ      CATR4
2226 012742 C05201      INC      R1
2227 012744 022701 000013      CMP      #13,R1
2228 012750 001371      BNE      CATR3
2229 012752 000000      HALT
2230 012754 006301      CATR4: ASL      R1          :FATAL ERROR MR. BOLSE!
2231 012756 005261 015200      INC      MINUSS(R1)      :MULTIPLY 'OFFSET' X2
2232 012762 000744      BR      CATR2
2233 012764 005267 002236      OVRHI: INC      ORHIGH
2234 012770 000741      BR      CATR2
2235 012772 005267 002200      OVRLO: INC      ORLOW
2236 012776 000736      BR      CATR2

```

```

2237
2238
2239
2240 013000 016767 002206 002222
2241 013006 066767 002202 002214
2242 013014 066767 002170 002206
2243 013022 016767 002202 002202
2244 013030 066767 002162 002174
2245 013036 066767 002144 002166
2246 013044 016767 002162 002162
2247 013052 066767 002142 002154
2248 013060 066767 002120 002146
2249 013066 016767 002142 002142
2250 013074 066767 002122 002134
2251 013102 066767 002074 002126
2252 013110 000002
2253
2254
2255
2256
2257 013112 005077 001730
2258 013116 017605 000000
2259 013122 062716 000002
2260 013126 012767 000006 001776
2261 013134 012767 000376 000064
2262 013142 000401
2263 013144 006115
2264 013146 006115
2265 013150 006115
2266 013152 111567 001756
2267 013156 146767 000044 001750
2268 013164 052767 000260 001742
2269 013172 132777 000200 001654
2270 013200 100374
2271 013202 116777 001726 001646
2272 013210 012767 000370 000010
2273 013216 005367 001710
2274 013222 001350
2275 013224 000002
2276 013226 000376
2277
2278
2279
2280 013230 105777 001614
2281 013234 100001
2282 013236 104011
2283 013240 000002

```

;ADD THE COUNTS AND SAVE TOTAL IN SPREADS OF '1-4'

```

CATRS:  MOV    AVGCNT,XSPRD1
        ADD    PLUS1,XSPRD1
        ADD    MINUS1,XSPRD1    ;=TO NO. COUNTS AT SPREAD OF '1'
        MOV    XSPRD1,XSPRD2
        ADD    PLUS2,XSPRD2
        ADD    MINUS2,XSPRD2    ;=TO NO. COUNTS AT SPREAD OF '2'
        MOV    XSPRD2,XSPRD3
        ADD    PLUS3,XSPRD3
        ADD    MINUS3,XSPRD3    ;=TO NO. COUNTS AT SPREAD OF '3'
        MOV    XSPRD3,XSPRD4
        ADD    PLUS4,XSPRD4
        ADD    MINUS4,XSPRD4    ;=TO NO. COUNTS AT SPREAD OF '4'
        RTI    ;EXIT

```

;SUBROUTINE TO TYPEOUT A '6' DIGIT OCTAL NO. THE 'PC' CONTAINS
;THE ADDRESS OF 'WORD' TO BE TYPED

```

OCTPRT: CLR    @PSW    ;ENABLE KEYBOARD INTR.
        MOV    @SP,R5    ;THE ADDRESS OF WORD TO BE TYPED
        ADD    #2,@SP    ;SET UP STACK TO EXIT
        MOV    #6,TEMP1
        MOV    #376,MASK    ;MASK FOR FIRST BIT
        BR    .+4
SHIFT:  ROL    (R5)
        ROL    (R5)
        ROL    (R5)
        MOVB   (R5),TEMP2
        BICB   MASK,TEMP2
        BIS    #260,TEMP2
        BITB   #200,@TPS
        BPL    .-6    ;WAIT FOR PRINTER READY
        MOVB  TEMP2,@TPB    ;PRINT CHAR.
        MOV    #370,MASK    ;MASK FOR NEXT '5' DIGITS
        DEC   TEMP1
        BNE   SHIFT
        RTI
MASK:   376

```

;SUBROUTINE TO TEST FOR THE KEYBOARD FLAG BEING SET

```

TKSFLG: TSTB   @TKS    ;FLAG SET?
        BPL    .+4    ;NO, EXIT
        TTYIN
        RTI    ;YES, INQUIRE

```


2325				
2326				
2327	013422	000		
2328	013423	045	042101	031117
2329	013430	040457	030504	020061
2330	013436	044504	043501	047516
2331	013444	052123	041511	052040
2332	013452	051505	026124	024040
2333	013460	040515	047111	042504
2334	013466	026503	030461	042055
2335	013474	040532	040504	041455
2336	013502	040051		
2337				
2338	013504	040445	042057	046040
2339	013512	047105	052107	037510
2340	013520	040040		
2341				
2342	013522	021045	507	047111
2343	013530	04044	1503	051125
2344	013536	0415	20131	042524
2345	013544	0521	027042	051440
2346	013552	050125	046120	020131
2347	013560	044124	020105	047506
2348	013566	046114	053517	047111
2349	013574	020107	047526	052114
2350	013602	043501	051505	022440
2351	013610	052040	020117	042523
2352	013616	042514	052103	042105
2353	013624	041440	027110	020054
2354	013632	054524	042520	023440
2355	013640	051103	020047	047524
2356	013646	051440	040524	052122
2357	013654	052040	051505	027124
2358	013662	100		
2359				
2360	013663	045	054524	042520
2361	013670	046040	052105	042524
2362	013676	020122	020047	020047
2363	013704	047524	051040	047125
2364	013712	042040	051535	051111
2365	013720	042105	052040	051505
2366	013726	035124	045	
2367	013731	047	023514	043517
2368	013736	041511	020054	041447
2369	013744	040447	044514	051102
2370	013752	052101	047511	026116
2371	013760	037440	023522	050105
2372	013766	050505	044524	044502
2373	013774	044514	054524	020054
2374	014002	043447	040447	047111
2375	014010	020054		
2376	014012	023522	023505	047503
2377	014020	042526	054522	020054
2378	014026	052047	047047	047116
2379	014034	040045		
2380				

:MESSAGES

TITLE: .BYTE
.ASCII '%AD02/AD11 DIAGNOSTIC TEST, (MAINDEC-11-DZADA-C)@'

MES2: .ASCII '%A/D LENGTH' @'

MES3: .ASCII '%GAIN ACCURACY TEST'. SUPPLY THE FOLLOWING VOLTAGES %'

.ASCII " TO SELECTED CH., TYPE 'CR' TO START TEST.@"

MES4: .ASCII "%TYPE LETTER ' ' TO RUN DESIRED TEST:%"

.ASCII "'L'OGIC, 'C'ALIBRATION, 'R'EPEATIBILITY, 'G'AIN. "

.ASCII "'R'E'COVERY, 'T'MIN@"

2381	014036	041536	040045		CNTRLC: .ASCII	'↑C%'
2382						
2383	014042	040536	100		CNTRLA: .ASCII	'↑A'
2384						
2385	014045	045	100		CRLF: .ASCII	'%'
2386						
2387	014047	045	040056		DOT: .ASCII	'%. '
2388						
2389	014052	020077	100		QMARK: .ASCII	'? '
2390						
2391	014055	045	046042	043517	MESS: .ASCII	'%LOGIC TEST%'
2392	014062	041511	052040	051505		
2393	014070	021124	040045			
2394						
2395	014074	044502	020124	020062	MES6: .ASCII	'BIT 2 SET INTR. LEVEL '
2396	014102	042523	020124	047111		
2397	014110	051124	020056	042514		
2398	014116	042526	020114	100		
2399						
2400	014123	045	041442	046101	MES7: .ASCII	'%CALIBRATION TEST%'
2401	014130	041111	040522	044524		
2402	014136	047117	052040	051505		
2403	014144	021124	040045			
2404	014150	021045	042522	047503	MES8: .ASCII	'%RECOVERY TEST'
2405	014156	042526	054522	052040		
2406	014164	051505	021124	100		
2407						
2408	014171	045	044103	020056	MES9: .ASCII	'%CH. '
2409	014176	100				
2410						
2411	014177	045	042447	054047	MES10: .ASCII	'%E'XT. OR 'I'NT. SYNC? '
2412	014204	027124	047440	020122		
2413	014212	044447	047047	027124		
2414	014220	051440	047131	037503		
2415	014226	040040				
2416						
2417						
2418	014230	051105	047522	020122	MES12: .ASCII	'ERROR BIT SET!%'
2419	014236	044502	020124	042523		
2420	014244	020524	040045			
2421						
2422	014250	021045	042522	042520	MES13: .ASCII	'%REPEATIBILITY TEST'
2423	014256	052101	041111	046111		
2424	014264	052111	020131	042524		
2425	014272	052123	040042			
2426						
2427	014276	022445	044103	024056	MES14: .ASCII	'%CH.(S)? '
2428	014304	024523	020077	100		
2429						
2430						
2431	014311	103	052517	052116	MES16: .ASCII	'COUNT SPREAD? '
2432	014316	051440	051120	040505		
2433	014324	037504	040040			
2434						
2435	014330	040507	047111	051450	MES18: .ASCII	'GAIN(S)? '
2436	014336	037451	040040			

.EVEN

; REGISTER ADDRESS

2526					
2527					
2528					
2529					
2530	015046	177776	PSW:	177776	; ADDRESS OF PROCESSOR STATUS REG
2531	015050	177560	TKS:	177560	; ADDRESS OF KEYBOARD STATUS REG
2532	015052	177562	TKB:	177562	; ADDRESS OF KEYBOARD BUFFER REG.
2533	015054	177564	TPS:	177564	; ADDRESS OF PRINTER STATUS REG.
2534	015056	177566	TPB:	177566	; ADDRESS OF PRINTER BUFFER REG
2535	015060	177570	SWR:	177570	; ADDRESS OF SWITCH REG.
2536	015062	177571	SWRO:	177571	; ADDRESS OF SWITCH REG. HIGH BYTE.
2537	015064	176770	ADCS:	176770	; A/D STATUS REG ADDRESS
2538	015066	176771	ADCSO:	176771	; HIGH BYTE OF A/D STATUS
2539	015070	176772	ADDBR:	176772	; A/D BUFFER REG.
2540	015072	000130	ADINT:	130	; A/D VECTOR ADDRESS

; ADDRESS AND CONSTANTS TABLE

2541					
2542					
2543					
2544	015074	000000	ADWORD:	0	; LOW BYTE OF 'ADWORD'
2545	015076	000000	ADSIGN:	0	; BIPOLAR=0, UNIPOLAR=1
2546	015100	000000	SUBX:	0	; LOGIC TEST SUBROUTINE SWITCH
2547	015102	000000	INTFLG:	0	
2548	015104	177634	DELAY1:	-100.	; DELAY COUNT FOR LOGIC TEST
2549	015106	001000	STACK:	1000	; INITIAL SP. ADDRESS
2550	015110	001262	AVECTOR:	INITA	; 'IA' VECTOR ADDRESS
2551	015112	000000	ADSIZE:	0	; OCTAL STORAGE OF A/D LENGTH
2552	015114	000000	PROC:	0	; TEMP STORAGE FOR 'PSW'
2553	015116	000000	CHRCNT:	0	; TEMP STORAGE
2554	015120	000000	COUNT:	0	; TEMP STORAGE
2555	015122	000000	KSTOR1:	0	; PERMANENT STORAGE
2556	015124	000000	KSTOR2:	0	; PERMANENT STORAGE
2557	015126	000000	KSTOR3:	0	; PERMANENT STORAGE
2558	015130	000000	KSTOR4:	0	; PERMANENT STORAGE
2559	015132	000000	TEMP1:	0	; TEMPORARY STORAGE
2560	015134	000000	TEMP2:	0	; TEMPORARY STORAGE
2561	015136	000000	TEMP3:	0	; TEMPORARY STORAGE
2562	015140	000000	MESPRT:	0	
2563	015142	000000	HIORDV:	0	
2564	015144	000000	HIGH:	0	
2565	015146	000000	LOW:	0	
2566	015150	000000	AVERM5:	0	
2567	015152	000000	AVERM4:	0	
2568	015154	000000	AVERM3:	0	
2569	015156	000000	AVERM2:	0	
2570	015160	000000	AVERM1:	0	
2571	015162	000000	AVRAGE:	0	
2572	015164	000000	AVERP1:	0	
2573	015166	000000	AVERP2:	0	
2574	015170	000000	AVERP3:	0	
2575	015172	000000	AVERP4:	0	
2576	015174	000000	AVERP5:	0	
2577	015176	000000	ORLOW:	0	
2578	015200	000000	MINUS5:	0	
2579	015202	000000	MINUS4:	0	
2580	015204	000000	MINUS3:	0	
2581	015206	000000	MINUS2:	0	

2582	015210	000000
2583	015212	000000
2584	015214	000000
2585	015216	000000
2586	015220	000000
2587	015222	000000
2588	015224	000000
2589	015226	000000
2590	015230	000000
2591	015232	000000
2592	015234	000000
2593	015236	000000
2594	015240	000000
2595	015242	000000
2596	015244	000000
2597	015246	000000
2598	015250	000000
2599	015252	000000
2600	015254	000000
2601	015256	000000
2602	015260	000000
2603	015262	000000
2604	015264	000000
2605	015266	000000
2606	015270	000000
2607	015272	000000
2608	015274	000000
2609	015276	000000
2610	015300	000000
2611	015302	000000
2612	015304	000000
2613	015306	000000
2614	015310	000000
2615	015312	000000
2616	015314	000000
2617	015316	000000
2618	015320	000000
2619	015322	000000
2620	015324	000000
2621	015326	000000
2622	015330	000000
2623	015332	000000
2624	015334	000000
2625	015336	000000
2626	015340	000000
2627	015342	000000
2628	015344	000000
2629	015346	000000
2630	015350	000000
2631	015352	000000
2632	015354	000000
2633	015356	000000
2634	015360	000000
2635	015362	000000
2636	015364	000000
2637	015366	000000

MINUS1:	0
AVGCNT:	00
PLUS1:	00
PLUS2:	00
PLUS3:	00
PLUS4:	00
PLUS5:	00
ORHIGH:	00
XSPRD1:	00
XSPRD2:	00
XSPRD3:	00
XSPRD4:	00
POS500:	00
POS250:	00
POS125:	00
POS625:	00
POS312:	00
NEG500:	00
NEG250:	00
NEG125:	00
NEG625:	00
NEG312:	00
GP50X1:	00
GP25X1:	00
GP12X1:	00
GP62X1:	00
GP31X1:	00
GP25X2:	00
GP12X2:	00
GP62X2:	00
GP31X2:	00
GP15X2:	00
GP12X4:	00
GP62X4:	00
GP31X4:	00
GP15X4:	00
GP07X4:	00
GP62X8:	00
GP31X8:	00
GP15X8:	00
GP07X8:	00
GP03X8:	00
GM50X1:	00
GM25X1:	00
GM12X1:	00
GM62X1:	00
GM31X1:	00
GM25X2:	00
GM12X2:	00
GM62X2:	00
GM31X2:	00
GM15X2:	00
GM12X4:	00
GM62X4:	00
GM31X4:	00
GM15X4:	00

2638	015370	000000
2639	015372	000000
2640	015374	000000
2641	015376	000000
2642	015400	000000
2643	015402	000000
2644		
2645		
2646	015400	015400
2647	015400	000000
2648		
2649		
2650		001000

GM07X4: 0
 GM62X8: 0
 GM31X8: 0
 GM15X8: 0
 GM07X8: 0
 GM03X8: 0

:HERE STARTS THE '512' WORD A/D DATA BUFFER.

ADBUFF: 0
 .=15400

.END INIT

015402	1703	2643									
015370	1679	2638									
015400	1685	2642									
015340	1496	2626									
015350	1502	2630									
015360	1507	2634									
015356	1643	2633									
015366	1649	2637									
015376	1655	2641									
015336	1460	2625									
015346	1467	2629									
015344	1595	2628									
015354	1601	2632									
015364	1607	2636									
015374	1613	2640									
015334	1437	2624									
015343	1542	2627									
015352	1547	2631									
015362	1553	2635									
015372	1559	2639									
015332	1693	2623									
015320	1663	2618									
015338	1669	2622									
015270	1475	2606									
015300	1481	2610									
015310	1486	2614									
015306	1621	2613									
015316	1627	2617									
015326	1633	2621									
015266	1445	2605									
015276	1451	2609									
015274	1567	2608									
015304	1573	2612									
015314	1579	2616									
015324	1585	2620									
015264	1427	1707	1743	2604							
015272	1515	2607									
015302	1520	2611									
015312	1526	2615									
015322	1532	2619									
017066	1411	1414	1732	1735							
007110	1422										
010022	1712	1738									
010114	1731	1733									
007146	1429	1440									
007230	1453	1470									
007326	1488	1510									
007444	1534	1562									
007562	1587	1616									
007660	1635	1658									
007736	1671	1688									
007774	1695	1707									
000000	1512	1425	1435	1443	1458	1473	1494	1513	1540	1565	1593
000010	1513	1449	1465	1479	1500	1518	1545	1571	1599	1619	1641
000020	1514	1484	1505	1524	1551	1577	1605	1625	1647	1661	1677
000030	1515	1530	1557	1583	1611	1631	1653	1667	1683	1691	1701

TEST16	002630	918#				
TEST17	002700	934#				
TEST18	002730	947#				
TEST19	003030	967#				
TEST2	001717	722#				
TEST20	003126	987#				
TEST21	003204	1030#				
TEST22	003276	1032#				
TEST23	003370	1034#				
TEST24	003462	1036#				
TEST25	003554	1038#				
TEST26	003646	1040#				
TEST27	003740	1046#				
TEST28	004024	1065#				
TEST29	004126	1126#				
TEST3	001740	733#				
TEST30	004300	1126	1128#			
TEST31	004452	1129	1130#			
TEST32	004624	1130	1132#			
TEST33	004776	1132	1134#			
TEST34	005150	1134	1136#			
TEST35	005322	1136	1138#			
TEST36	005474	1138	1140#			
TEST37	C 3600	1162#				
TEST38	005716	1172	1184#			
TEST39	005752	1198#				
TEST4	002022	752#				
TEST5	002056	764#				
TEST6	002114	776#				
TEST7	002152	789#				
TEST8	002214	803#				
TEST9	002252	815#				
TINT0	003014	952	962#			
TINTE	003106	969	980#			
TINTX	004112	1072	1081#			
TINT08	005416	1138#				
TINT1A	003724	1040#				
TINT1B	005244	1136#				
TINT2	006044	1204	1214#			
TINT2A	003632	1038#				
TINT2B	005072	1134#				
TINT3	006070	1213	1215	1219	1220#	
TINT3A	003540	1036#				
TINT3B	004720	1132#				
TINT4A	003446	1034#				
TINT4B	004546	1130#				
TINT5A	003354	1032#				
TINT5B	004374	1128#				
TINT6A	003262	1030#				
TINT6B	004222	1126#				
TINT7A	003164	990	997#			
TINT7B	004006	1053	1058#			
TITLE	013423	582	2328#			
TYB	015052	1896	2532#			
TYC	015050	580*	634*	639*	1894	2280
TYFLG	013230	2091	2280#			2531#

AD02 AD11 DIAGNOSTIC MACY11 27(732) 16-SEP-76 16:37 PAGE 76
DZADAC.P11 CROSS REFERENCE TABLE -- MACRO NAMES

INT	1007*	1029	1031	1033	1035	1037	1039	
INT2	1086*	1125	1127	1129	1131	1133	1135	1137

ADC	2174	2182													
ADD	1744	1759	1761	1763	1862	1873	1972	1973	1974	2071	2101	2128	2133	2164	2172
	2173	2189	2199	2241	2242	2244	2245	2247	2248	2250	2251	2253			
ASL	622	1865	1866	1867	1969	1970	1971	2069	2230						
ASR	613	619	1463	2178											
BEQ	578	605	632	657	682	689	697	1179	1233	1290	1317	1357	1384	1429	1453
	1488	1534	1587	1635	1671	1695	1771	1773	1775	1777	1864	1928	1938	1961	2108
	2138	2216	2225												
BGE	1394														
BGT	1886	1965	2005	2170											
BIC	695	831	845	864	907	922	954	973	1074	1150	1170	1206	1269	1861	1897
	1966	2070	2292												
BIOB	798	1267	2267												
BIS	692	951	970	977	989	1030	1032	1034	1036	1038	1040	1047	1066	1078	1126
	1128	1130	1132	1134	1136	1138	1148	1153	1167	1203	1273	1382	1783	2142	2268
BISB	779	820													
BIT	688	1334	1336	1776	1779	1844	2287	2309	2311						
BITB	2269														
BLE	1958	2167													
BLT	1963														
BMI	716	730	761	773	800	812	837	851	871	883	912	928	1319	1899	1901
	1909	2219	2221												
BNE	589	593	597	601	615	621	645	648	651	654	660	714	739	747	759
	771	784	797	810	823	1126	1128	1130	1132	1134	1136	1138	1182	1272	1335
	1337	1339	1341	1344	1346	1350	1352	1355	1361	1379	1391	1731	1734	1748	1755
	1780	1782	1787	1792	1797	1845	1903	1917	1923	1926	1934	1940	1979	2041	2045
	2049	2053	2067	2105	2131	2136	2176	2181	2195	2205	2212	2228	2274	2288	2310
	2312														
BPL	679	708	727	741	749	786	825	854	868	880	886	894	898	901	915
	931	941	944	958	985	1000	1030	1032	1034	1036	1038	1040	1052	1071	1125
	1128	1130	1132	1134	1136	1138	1145	1176	1189	1193	1196	1210	1228	1276	1287
	1871	1883	1895	1911	1998	2000	2103	2113	2127	2144	2270	2281	2302	2314	
BR	591	595	599	603	608	625	664	961	979	996	1030	1032	1034	1036	1038
	1040	1057	1080	1126	1128	1130	1132	1134	1136	1138	1154	1172	1213	1219	1254
	1281	1291	1342	1347	1353	1395	1738	1790	1795	1800	1811	1853	1868	1905	1913
	1932	1936	1946	1975	2110	2116	2134	2140	2232	2234	2236	2262			
CLC	1264														
CLP	583	640	670	687	756	781	821	834	835	848	881	925	942	965	983
	1002	1030	1032	1034	1036	1038	1040	1050	1062	1069	1094	1126	1128	1130	1132
	1134	1136	1138	1143	1158	1165	1216	1224	1309	1419	1860	1874	1887	1893	1954
	1955	1956	2099	2120	2159	2210	2222	2257	2319						
CLRE	705	728	736	744	755	767	768	793	818	832	847	924	938	950	963
	972	981	992	995	1030	1032	1034	1036	1038	1040	1049	1060	1068	1077	1091
	1126	1128	1130	1132	1134	1136	1138	1142	1159	1164	1202	1222	2042		
CMP	588	592	596	600	604	614	620	659	696	962	980	998	1059	1082	1126
	1128	1130	1132	1134	1136	1138	1156	1178	1214	1221	1271	1289	1318	1338	1340
	1343	1345	1349	1351	1354	1356	1383	1390	1393	1770	1772	1774	1870	1908	1964
	2040	2044	2048	2052	2137	2166	2169	2211	2218	2220	2224	2227	2313	2316	
CMPB	644	647	650	653	1791	1796	1898	1900	1902	1916	1922	1925	1933	1937	1939
	1960	1962	2104	2107											
DEC	1181	1266	1733	1747	1754	1863	1885	1929	1959	2004	2175	2180	2194	2201	2204
	2215	2273													
EMT	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567
	568														
HALT	534	2018	2068	2229	2303										
INC	113	738	746	758	770	783	796	807	809	849	910	1191	1392	1907	2125

M06

AD02 AD11 DIAGNOSTIC MACY11 27(732) 16-SEP-76 16:37 PAGE 80
DZADAC.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DZADAC/SOL/CRF/PAGNUM=DZADAC
RUN-TIME: 9 17 3 SECONDS
RUN-TIME RATIO: 85/30=2.7
CORE USED: 10K (19 PAGES)

10			...B1	2103	012062	100375	...B5
420			...C1	2156			...C5
97			...D1	2194	012570	005367	...D5
155			...E1	2246	013044	016767	...E5
213			...F1	2293	013276	005777	...F5
267			...G1	2335	013466	026503	...G5
318			...H1	2390	014052	020077	...H5
372			...I1	2445	014410	020064	...I5
431			...J1	2502	014734	100	...J5
491			...K1	2525	015060	177570	...K5
			...L1	2591	015232	000000	...L5
			...M1	2647	015400	000000	...M5
		002000	...N1				...N5
			...B2	CHRCNT	015116		...B6
514	000034	013242	...C2	GM25X1	015336		...C6
573	001013	001402	...D2	INPLTA	011040		...D6
635	001324	104000	...E2	*INUS1	015210		...E6
674			...F2	PWRUP	011570		...F6
708	001546	100001	...G2				...G6
761	002052	100401	...H2	TEST24	003462		...H6
812	002246	100401	...I2	TSTTKS=	104016		...I6
866	002442	012711	...J2				...J6
912	002612	100401	...K2	BGT	1886	1965	...K6
956	002772	100401	...L2		982	990	...L6
991	003162	003402	...M2				...M6
			...N2	**END**	USER DAVIES, TOM		...N6
			...B3				
			...C3				
1050	003756	005011	...D3				
1094			...E3				
1133	004776		...F3				
1189	005732	100375	...G3				
1207	006016	012711	...H3				
1249	006164	012767	...I3				
1301			...J3				
1352	006664	001011	...K3				
1405			...L3				
1456	007202	014521	...M3				
			...N3				
1507	007324	015360	...B4				
1558	007440	015252	...C4				
1612	007556	015254	...D4				
1666	007674	104013	...E4				
1713	010024	014765	...F4				
1766	010260	004767	...G4				
1822	010474	104000	...H4				
1863	010712	005367	...I4				
1899	011062	100427	...J4				
1953	011312	012703	...K4				
1996	011504	100375	...L4				
2042	011654	105067	...M4				
2066	011754	005716	...N4				